

ELISA MANNES

**UM ESQUEMA BIO-INSPIRADO PARA TOLERÂNCIA À
MÁ-CONDUTA EM SISTEMAS DE QUÓRUM APOIANDO
SERVIÇOS DE OPERAÇÃO EM MANETS**

Dissertação Apresentada como Requisito Parcial
à Obtenção do Grau de Mestre. Programa de
Pós-Graduação em Informática, Setor de Ciências
Exatas, Universidade Federal do Paraná.

Orientador: Prof. Aldri Luiz dos Santos
Coorientadora: Profa. Michele Nogueira Lima

CURITIBA

2011

ELISA MANNES

**UM ESQUEMA BIO-INSPIRADO PARA TOLERÂNCIA À
MÁ-CONDUTA EM SISTEMAS DE QUÓRUM APOIANDO
SERVIÇOS DE OPERAÇÃO EM MANETS**

Dissertação Apresentada como Requisito Parcial
à Obtenção do Grau de Mestre. Programa de
Pós-Graduação em Informática, Setor de Ciências
Exatas, Universidade Federal do Paraná.

Orientador: Prof. Aldri Luiz dos Santos
Coorientadora: Profa. Michele Nogueira Lima

CURITIBA

2011

Mannes, Elisa

Um esquema bio-inspirado para tolerância à má-conduta em sistemas de quórum apoiando serviços de operação em MANETs / Elisa Mannes. – Curitiba, 2011.

81 f.: il., tab.

Dissertação (mestrado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática.

Orientador: Aldri Luiz dos Santos

Coorientadora: Michele Nogueira Lima

1. Redes de computadores – Medidas de segurança.

2. Sistemas de computação sem fio - Medidas de segurança.

I. Santos, Aldri Luiz dos. II. Lima, Michele

Nogueira. III. Universidade Federal do Paraná. IV. Título.

CDD: 004.6



Ministério da Educação
Universidade Federal do Paraná
Programa de Pós-Graduação em Informática

PARECER


Nós, abaixo assinados, membros da Banca Examinadora da defesa de Dissertação de Mestrado em Informática, da aluna Elisa Mannes, avaliamos o trabalho intitulado, "UM ESQUEMA BIO-INSPIRADO PARA TOLERÂNCIA À MÁ-CONDUTA EM SISTEMAS DE QUÓRUM APOIANDO SERVIÇOS DE OPERAÇÃO EM MANETs", cuja defesa foi realizada no dia 16 de dezembro de 2011, às 09:30 horas, no Departamento de informática do Setor de Ciências Exatas da Universidade Federal do Paraná. Após a avaliação, decidimos pela aprovação da candidata.

Curitiba, 16 de dezembro de 2011.


Prof. Dr. Aldri Luiz dos Santos
DINF/UFPR – Orientador


Prof. Dr. Michele Nogueira Lima
DINF/UFPR - Coorientadora


Prof. Dr. Fabíola Gonçalves Pereira Greve
DCC/UFBA – Membro Externo


Prof. Dr. Carmem Satie Hara,
DINF/UFPR – Membro Interno



DEDICATÓRIA

Este trabalho é dedicado a minha família, aos amigos do grupo de pesquisa NR2 e a comunidade acadêmica.

AGRADECIMENTOS

Agradeço a Deus pela minha vida e minha saúde, pela proteção e discernimento. Agradeço ao meu noivo, Luiz Carlos, pelo imenso carinho e pela atenção comigo e com a minha vida acadêmica. Obrigada pelas ideias inteligentes e pelas diversas discussões do texto, solução e artigos. Sua ajuda foi essencial, te amo muito. Agradeço aos meus irmãos Mariana e Elton, em especial ao Elton pelas infinitas caronas e pela companhia enquanto eu estava em Curitiba. Valeu! Aos meus pais, Orlando e Marisa, muito obrigada pelo apoio sempre.

Agradeço também aos meus amigos do grupo de pesquisa NR2: Cinara, Neimar, Larissa, Crystiane, Fernando, Marwin, Juliana, Juliano, Guilherme, Nadine, Robinho, Julio, Eduardo, Marco e Urlan, e aos agregados Emanuel, Rebeca, Gisane, Rafa, Leila, Srishti e Jenny, pela companhia diária nesses dois anos. Me divirto muito com vocês, vocês fizeram a diferença durante o mestrado, obrigada! Um agradecimento especial para a Cinara e a Rebeca, pela companhia no hotel e nas viagens de Joinville à Curitiba. Meninas, obrigada pelas infinitas risadas, histórias e piadas. Os *check-ins* no hotel e os congestionamentos na BR não seriam os mesmos sem vocês :)

Meu obrigado também à CAPES, pelo financiamento dos meus estudos nesses dois anos através do Programa de Fomento à Pós-Graduação (PROF), e ao Departamento de Informática da UFPR e funcionários, pelo apoio acadêmico. Agradeço de forma especial aos orientadores Aldri e Michele, pela atenção e orientação dada à minha dissertação, e pela preocupação com a minha formação acadêmica e profissional. Obrigada também aos professores Elias, Carmem e Albini, pelas lições nas disciplinas cursadas. Agradeço também às professoras Carmem (UFPR) e Fabíola (UFBA), pela participação na banca de defesa da minha dissertação e pelos valiosos comentários. Obrigado também ao professor Wagner, da Estatística e ao professor André, da Informática, pela ajuda nas dúvidas surgidas durante a confecção dessa dissertação.

Muito obrigada!

Oh, yes, I am very odd - that is to say, I am
methodical, orderly, and logical, and I do not
like to distort facts to support a theory.

Agatha Christie's Hercule Poirot

RESUMO

As redes *ad hoc* móvel (MANETs) são formadas dinamicamente por dispositivos móveis (nós) com restrição de recursos. Os serviços de operação de rede precisam lidar com as características dessas redes, como a mobilidade e a falta de recursos dos nós, a fim de gerenciar os seus dados e de apoiar o funcionamento das aplicações. A disponibilidade de dados geralmente é obtida por técnicas de replicação, sendo que os sistemas de quórum têm se apresentado como um método eficaz na replicação de dados em MANETs, provendo robustez às aplicações. Porém, os sistemas de quórum existentes para as MANETs não focam na segurança dos dados e das operações, sendo vulneráveis à ação de nós de má-conduta. A maioria dos mecanismos atuais para a detecção de nós de má-conduta em MANETs utiliza entidades centrais ou necessita da confiança entre os nós para uma correta detecção, o que resulta em uma sobrecarga de mensagens. Desta forma, este trabalho propõe um esquema para a tolerância de nós de má-conduta nas operações de replicação em um sistema de quórum probabilístico. O esquema proposto, chamado de QS^2 , tem como inspiração os mecanismos biológicos de sensoriamento em quórum e de seleção por parentesco, ambos encontrados em bactérias. Diferentemente dos sistemas existentes na literatura, o QS^2 é autônomo, auto-organizado e distribuído. Nesse esquema, os nós monitoram a qualidade da interação entre eles e classificam e selecionam os nós de acordo com o comportamento observado. O QS^2 é avaliado por meio de simulações e os resultados obtidos mostram que, comparado com um sistema de quórum probabilístico para MANETs sem o uso do QS^2 , ele proporciona um aumento de até 87% na confiabilidade em cenários com ataques de injeção de dados nas operações de replicação. Além disso, o esquema apresenta uma eficácia na detecção de nós egoístas em torno de 98,5% com uma taxa de falsos positivos menor que 2%, enquanto que na identificação de nós maliciosos a eficácia é em média de 80%, com uma taxa de falsos positivos inferior a 1%. O QS^2 foi aplicado e avaliado em dois cenários realísticos de MANETs, e nesses cenários ele proporcionou uma melhora superior a 55% na confiabilidade dos dados, observando que nesses cenários a constante mudança de topologia resultou em uma quantidade de dados desatualizados superior a quantidade de dados falsos no sistema de replicação.

Palavras-chave: MANETs, sistemas de quórum, serviços de operação da rede, segurança.

ABSTRACT

Mobile Ad Hoc Networks (MANETs) consist of mobile devices (nodes) which dynamically exchange data without any fixed base station. Operational services have to deal with peculiar characteristics of these networks, such as mobility and lack of resources, in order to manage their data and to support applications. The availability of data is usually obtained by replication, and quorum systems have been used as an effective method for data replication in MANETs, providing robustness to applications. However, existing quorum systems for MANETs do not focus on data security and operations are vulnerable to the action of misbehaving nodes. Most of the current mechanisms for misbehavior detection use central entities or need to establish trust relationships among nodes to correctly detect misbehaving nodes, resulting in an overhead to the system. Thus, this work proposes a scheme to tolerate misbehaving nodes in replication operations on a probabilistic quorum system for MANETS. The proposed scheme, called QS^2 , has been inspired by biological mechanisms in quorum sensing and kin selection, both present in bacteria. Unlike existing systems in the literature, QS^2 is autonomous, self-organized and distributed. In this scheme, nodes monitor the quality of interaction and select and classify other nodes according to the observed behavior. QS^2 is evaluated through simulations and the results show that, compared with a probabilistic quorum system for MANETs without the use of QS^2 , it provides an increase of up to 87% in data reliability in scenarios with data injection attacks on data replication. Moreover, the scheme has a detection efficiency of selfish nodes about 98.5% with a false positive ratio of less than 2%, while the efficiency on the identification of malicious nodes is on average 80%, with a false positive ratio of less than 1%. QS^2 has been implemented and evaluated in two realistic scenarios for MANETs, and it provided an improvement higher than 55% in data reliability, observing that in these scenarios the constant change of topology resulted in a number of outdated data higher than the amount of false data in the replication system.

Keywords: MANETs, quorum system, network operation services, security.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	Problema	2
1.2	Objetivos	3
1.3	Contribuições	4
1.4	Estrutura da dissertação	5
2	GERÊNCIA DE DADOS DE OPERAÇÃO DE REDE EM MANETS	6
2.1	Sistemas de quórum	6
2.1.1	Quórums bizantinos	7
2.1.2	Quórums probabilísticos	8
2.1.3	Quórums probabilísticos para redes <i>ad hoc</i> (PAN)	9
2.2	Tipos de ataques no PAN	12
2.2.1	Falta de cooperação	12
2.2.2	Temporização	14
2.2.3	Injeção de dados	15
2.3	Avaliação do PAN diante de má-conduta	16
2.3.1	Ataque de falta de cooperação	18
2.3.2	Ataque de temporização	19
2.3.3	Ataque de injeção de dados	20
2.3.4	Participação de nós de má-conduta em quórums de leitura	21
2.4	Resumo	22
3	MECANISMOS DE TOLERÂNCIA À MÁ-CONDUTA EM MANETS	23
3.1	Deteção de nós egoístas e maliciosos	23
3.1.1	Sistemas de reputação	23
3.1.2	Identificação de injeção de dados falsos	24
3.2	Mecanismo bio-inspirado para tolerância a ataques	25
3.2.1	Sensoriamento em quórum (<i>Quorum Sensing</i>)	26
3.2.2	Seleção por parentesco (<i>Kin Selection</i>)	28
3.3	Resumo	29
4	UM ESQUEMA BIO-INSPIRADO PARA A TOLERÂNCIA DE NÓS DE MÁ-CONDUTA	30
4.1	Visão geral do esquema QS^2	30
4.1.1	Modelo do sistema	31
4.2	Entidades bio-inspiradas	34

4.3	Funcionamento do QS^2	36
4.3.1	Operações	37
4.3.2	Escrita	40
4.3.3	Leitura	40
4.4	Resumo	42
5	AVALIAÇÃO DO ESQUEMA QS^2	43
5.1	Cenários de validação	43
5.2	Métricas	45
5.3	Avaliação de desempenho	46
5.3.1	Ataque de falta de cooperação	47
5.3.2	Ataque de temporização	49
5.3.3	Ataque de injeção de dados	51
5.3.4	Nós egoístas e maliciosos	53
5.4	Avaliação de eficiência	54
5.4.1	Taxa de detecção	54
5.4.2	Taxa de falsos negativos	55
5.4.3	Taxa de falsos positivos	56
5.5	Resumo	58
6	SERVIÇOS DE OPERAÇÃO DE REDE CONFIÁVEIS	59
6.1	Serviços de operação de rede	59
6.2	Ambiente urbano - centro de uma cidade	60
6.2.1	Grau de confiabilidade	62
6.2.2	Eficiência	64
6.2.3	Dados falsos X dados desatualizados	64
6.3	Ambiente de transporte - linhas de ônibus	66
6.3.1	Grau de confiabilidade	68
6.3.2	Eficiência	69
6.3.3	Dados falsos X dados desatualizados	69
6.4	Resumo	71
7	CONSIDERAÇÕES FINAIS	72
	ANEXO	81

LISTA DE FIGURAS

2.1	Construção de um sistema de quórum \mathcal{Q}	7
2.2	Aplicação dos sistemas de quórum em redes e serviços	8
2.3	Operação de escrita no PAN	11
2.4	Operação de leitura no PAN	11
2.5	Ataque de falta de cooperação na leitura	13
2.6	Ataque de falta de cooperação na escrita	14
2.7	Ataque de temporização	14
2.8	Ataque de injeção de dados na leitura	15
2.9	Ataque de injeção de dados na escrita	16
2.10	G_c com ataque de falta de cooperação	18
2.11	G_c com ataque de temporização	19
2.12	G_c com ataque de injeção de dados	20
2.13	Q_r afetados por nós egoístas e maliciosos	21
3.1	Processo de sensoriamento em quórum e início de uma ação em conjunto	28
4.1	Arquitetura do esquema QS^2	31
4.2	Modelo do sistema em camadas	32
4.3	Disseminação e contagem de autoindutores no esquema QS^2	34
4.4	Informações enviadas pelo QS^2 nas operações de leitura e de escrita de dados	36
4.5	Monitoramento dos nós pelo QS^2	38
4.6	Classificação dos nós pelo QS^2	39
4.7	Decisão de cooperação pelo QS^2	40
5.1	Cenário de simulação	45
5.2	G_c do QS^2 sem ataque	47
5.3	G_c com ataque de falta de cooperação na escrita	48
5.4	G_c com ataque de falta de cooperação na leitura	48
5.5	G_c com ataque de temporização com $T=400s$	49
5.6	G_c com ataque de temporização com $T=800s$	50
5.7	G_c com ataque de temporização com $T=3000s$	51
5.8	G_c com ataque de injeção de dados na escrita	52
5.9	G_c com ataque de injeção de dados na leitura	52
5.10	G_c em uma MANET com nós egoístas e maliciosos	54
5.11	Tx_{det} de nós egoístas e maliciosos	55
5.12	Tx_{fn} na detecção de nós egoístas e maliciosos	56
5.13	Tx_{fp} na detecção de nós egoístas e maliciosos	57

5.14	Infográfico da frequência de detecção de nós maliciosos pelo QS^2	57
6.1	Cenário de simulação do centro de uma cidade [67]	61
6.2	G_c em cenários urbanos sem o uso do QS^2	63
6.3	G_c em cenários urbanos	63
6.4	Tx_{det} em cenários urbanos	64
6.5	Tx_{mis} e Tx_{out} em cenários com ataque de injeção de dados	65
6.6	Tx_{mis} e Tx_{out} em cenários com todos os ataques	65
6.7	Cenário de simulação de linhas de ônibus [69]	67
6.8	G_c em cenários de transporte sem o uso do QS^2	68
6.9	G_c em cenários de linhas de ônibus	69
6.10	Tx_{det} em cenários de linhas de ônibus	70
6.11	Tx_{mis} e Tx_{out} em cenários com ataque de injeção de dados	70
6.12	Tx_{mis} e Tx_{out} em cenários com todos os ataques	71

LISTA DE ALGORITMOS

1	Monitoramento dos nós pelo nó s_i	37
2	Classificação do nó s_w pelo nó s_i	38
3	Decisão de cooperação pelo nó s_i	39
4	Seleciona nós	40
5	Operação de escrita com o QS^2 realizada pelo nó s_i	41
6	Operação de leitura com o QS^2 realizada pelo nó s_i	41

LISTA DE TABELAS

2.1	Principais parâmetros de simulação dos cenários de avaliação	17
2.2	Síntese do impacto dos nós de má-conduta no PAN	22
4.1	Comportamento dos nós na rede	35
5.1	Principais parâmetros de simulação dos cenários de validação	44
5.2	Relação de velocidade máxima e tempo de pausa	53
5.3	Síntese do ganho na confiabilidade com o uso do QS^2	58
5.4	Síntese da eficiência do QS^2	58
6.1	Principais parâmetros de simulação dos cenários urbanos	62
6.2	Principais parâmetros de simulação dos cenários de transporte	67
6.3	Síntese do uso do QS^2 em ambientes realísticos de MANETs	71

LISTA DE ABREVIATURAS E SIGLAS

ACACIA	A Controller-node-based Access-Control mechanism for Ad hoc networks
AODV	Ad hoc On Demand distance Vector routing protocol
BBDS	Bad Behavior Detection System
CONFIDANT	Cooperation Of Nodes Fairness In Dynamic Ad-hoc NeTworks
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
MANET	Mobile Ad hoc NeTwork
MDQ	Mobile Dissemination Quorum system
NS-2	Network Simulator 2
PAN	Probabilistic quorum system for Ad hoc Network
QS^2	Quorum System + Quorum Sensing
SCAN	Self-organized network-layer seCurity in mobile Ad hoc Networks
STS	STorage Set

NOTAÇÃO

\mathcal{Q}	sistema de quórum
n	quantidade de nós em uma rede
s_n	identificação de um nó na rede
Q_w	quórum de escrita
Q_r	quórum de leitura
f	quantidade de nós de má-conduta no sistema
F	quantidade de nós para disseminar um dado (<i>fanout</i>)
v	dado replicado no sistema
T	intervalo de tempo entre a disseminação de dados
$AI-W$	autoindutor referente à escrita de dados
$AI-F$	autoindutor referente ao encaminhamento de dados
C	gene referente à um nó egoísta
M	gene referente à um nó malicioso
k_{env}	taxa de escritas por nó
k_{enc}	taxa de encaminhamentos por nó
λ	média de leituras e escritas por nó
k_{env}^{max}	taxa máxima de escritas
k_{enc}^{min}	taxa mínima de encaminhamentos
γ	probabilidade de envio de escritas ser menor que k_{env}^{max}
δ	probabilidade de envio de encaminhamentos ser maior que k_{enc}^{min}
$A(d, a)$	conjunto de interação de nós de má-conduta e resultado da detecção
$B(d, a)$	conjunto de interação de nós bons e resultado da detecção

CAPÍTULO 1

INTRODUÇÃO

Os avanços das tecnologias de comunicação sem fio reforçam o desenvolvimento e a utilização de diferentes redes sem fio em direção a criação das redes do futuro [1]. As redes *ad hoc* móveis (MANETs) fazem parte dessas redes e têm como objetivo dar suporte a diversas aplicações em áreas como saúde, transporte e entretenimento. As MANETs são redes de dispositivos (nós) móveis, formadas dinamicamente, em que os usuários participam da rede utilizando dispositivos móveis como *notebooks*, *smartphones* e *tablets*. Nelas, os serviços e as aplicações são fornecidos de uma forma distribuída e auto-organizada, e juntamente com outras redes sem fio, devem convergir a fim de suportar os serviços e as aplicações das redes do futuro.

Essas novas redes devem causar um impacto substancial na sociedade. Contudo, juntamente com as facilidades previstas com o uso de tais redes, surgem numerosos desafios e requisitos. Um desses desafios é a necessidade de oferecer aplicações continuamente apesar de falhas, acidentes, ataques ou qualquer outras condições adversas da rede [1]. Com a esperada complexidade das redes da próxima geração e a participação de diferentes entidades autônomas, surge a necessidade de projetar serviços confiáveis em todos os segmentos de rede, inclusive nas MANETs. As características das MANETs podem facilmente ocasionar a partição da rede, tornando os serviços indisponíveis e sustentando informações desatualizadas [2]. A falta de informação ou o uso de informações desatualizadas influenciam na operação dos nós, dos serviços e das aplicações, comprometendo o desempenho de toda a rede e até mesmo podendo causar a sua inutilização.

Nesse sentido, a gerência de dados é essencial para o fornecimento de serviços e aplicações confiáveis. Os serviços de operação da rede, tais como os serviços de localização de recursos e de gerenciamento da mobilidade, apoiam as aplicações através do monitoramento e do gerenciamento de dados de controle da rede. Esses serviços têm como principal função o envio de informações para os nós, para que eles sejam capazes de se antecipar e de se adaptar às situações adversas provenientes das próprias características das MANETs, como as constantes mudanças de topologia que provocam quebras de enlace e o consequente particionamento da rede. Por isso, é necessário que os serviços de operação de rede sejam seguros e robustos, com garantia de disponibilidade e de confiabilidade dos dados. Uma das formas comumente empregadas para prover a disponibilidade dos dados é por meio da redundância das informações, obtida através das técnicas de replicação dos dados [3].

A replicação de dados é uma técnica amplamente empregada na melhoria da dispo-

nibilidade dos dados em sistemas distribuídos [4, 5]. Porém, as abordagens clássicas de replicação de dados, como a replicação passiva [6] e a ativa [7] quando aplicadas em MANETs, apresentam um alto custo. Isso ocorre devido às restrições impostas por esses sistemas, como servidores estáticos e a garantia de entrega de mensagens, o que não reflete o ambiente das MANETs. Tais abordagens também não consideram a mobilidade e os recursos escassos dos nós. Embora existam técnicas de replicação que abordam a dinamicidade dos servidores [], enfatiza-se a replicação por sistema de quórum [8] como uma alternativa atraente, que busca um balanceamento da carga entre os servidores. Esse tipo de replicação considera subconjuntos de servidores para a realização de operações de replicação, diminuindo os custos de processamento e de comunicação. Por esta razão, um crescente interesse é observado na aplicação desses sistemas em serviços para MANETs [9, 10].

Dentre os sistemas de quórum para MANETs, aqueles formados com base em probabilidade diminuem as restrições impostas pelos sistemas de quórum tradicionais. Além disso, essa forma menos restrita de construir os quórums não é afetada pela mobilidade dos nós da rede, visto que os componentes dos quórums são escolhidos probabilisticamente. Por outro lado, esses sistemas não garantem uma consistência global dos dados e nem favorecem as operações de atualização de dados, sendo indicados para aplicações cuja garantia da disponibilidade é mais relevante que o custo de lidar com eventuais inconsistências [11]. Os serviços de operação da rede necessitam de alta disponibilidade dos dados nos nós, sendo que alguns deles suportam as inconsistências geradas pelo sistema de quórum, tornando os sistemas de quórum atrativos para a replicação de dados de controle da rede. Porém, o uso dos sistemas de quórums para a replicação dos dados precisa ser robusto e deve garantir a integridade e a disponibilidade dos dados replicados, de forma a suportar corretamente o funcionamento das aplicações.

1.1 Problema

Devido à forma como são organizadas, as MANETs estão sujeitas à ação de nós de má-conduta em suas operações. Porém, a ausência de uma entidade central de controle não permite que as MANETs implementem sistemas de segurança tradicionais, tais como *firewalls* e sistemas baseados em servidores confiáveis [12]. Além disso, a entrada e a saída constante de nós da rede afetam a confiabilidade das operações, uma vez que um nó de má-conduta pode entrar na rede e participar dos serviços de operação de rede [13]. Os sistemas de quórum existentes para MANETs não empregam mecanismos de segurança contra esses nós, o que representa uma vulnerabilidade no caso da participação de nós de má-conduta nas operações de replicação.

Estudos mostram que os sistemas de replicação são vulneráveis à ação de nós de má-conduta [14, 15], sendo que os sistemas de quórum para MANETs são vulneráveis

principalmente aos ataques de falta de cooperação, temporização e injeção de dados [16]. Esses ataques têm como consequência a perda de desempenho do sistema de replicação e comprometem a confiabilidade dos dados. Entretanto, o ataque de injeção de dados também leva o sistema a um estado inconsistente, que compromete a integridade dos dados e, consequentemente, o uso de tais dados pelas aplicações é prejudicado.

Apesar de existirem sistemas de replicação que são tolerantes a nós de má-conduta, como o PAXOS [17] e o sistema de quórum bizantino [8], tais sistemas sustentam fortes premissas, como a garantia de entrega das mensagens, o que é inviável nas MANETs devido às suas características. Ainda, o PAXOS oferece a tolerância a nós de má-conduta pela realização do consenso das operações, aplicado a replicação de máquina de estados, que não é aconselhável para as MANETs devido à sobrecarga resultante da quantidade de mensagens trocadas [18].

Uma técnica bastante utilizada para lidar com os nós de má-conduta nas MANETs é o controle da reputação dos nós [19]. Através de um esquema de reputação, um nó seleciona e interage somente com outros nós que apresentem bom comportamento, evitando a participação de nós de má-conduta e prejuízos à rede. Existem vários exemplos de esquemas de reputação em MANETs, tais como o CONFIDANT [20], SCAN [21] e o ACACIA [22]. Porém, a maioria desses sistemas baseiam-se em informações enviadas por uma terceira entidade que analisa o comportamento dos nós, tais como um BBDS (*Bad Behavior Detection System*) ou premissas de confiança consideradas na interação humana. Além disso, algumas soluções divulgam a recomendação sobre um nó para todos os outros nós da rede, gerando uma sobrecarga de mensagens. Outro problema encontrado nas soluções propostas é o fato de assumirem que os sistemas de detecção são livres de falhas e que sempre funcionam corretamente [23].

Desta forma, é necessário fornecer tolerância à má-conduta nos sistemas de quórum, assim como aplicar uma solução que seja autônoma, auto-organizada e de baixo custo contra os nós de má-conduta. Essas características podem ser naturalmente encontradas em muitos sistemas biológicos, sendo que os sistemas biológicos têm sido inspiração para o projeto de diversas soluções autônomas e auto-organizadas para as MANETs [24].

1.2 Objetivos

Diversos problemas na área da computação têm sido modelados e resolvidos com soluções inspiradas em comportamentos biológicos. Isso se deve às características encontradas na Biologia, tais como autonomia e auto-organização das entidades, que são também desejadas nos sistemas de computação. Este trabalho tem como objetivo propor um esquema que possa ser incorporado aos sistemas de quórum e que desta forma possibilite a exclusão de nós de má-conduta na participação das operações de replicação. Os mecanismos biológicos de sensoramento em quórum e de seleção por parentesco, ambos encontrados em

bactérias, apresentam características de auto-organização e autonomia, evitando a participação de entidades maliciosas. O sensoramento em quórum determina a densidade de bactérias no ambiente e iniciam ações orientadas a eventos, enquanto que a seleção por parentesco promove a cooperação entre as bactérias que compartilham o mesmo material genético. Juntos, esses mecanismos evitam que bactérias mutantes interfiram no funcionamento do sensoramento, além de dificultar a reprodução de tais bactérias.

Os mecanismos biológicos de sensoramento em quórum e de seleção por parentesco apresentam diversas semelhanças com o comportamento dos nós em uma MANET. Uma delas é o fato de que cada bactéria monitora individualmente a concentração de outras bactérias no ambiente e realiza ações de forma autônoma. Esse comportamento é adequado para as MANETs, pois os nós são independentes e não possuem o apoio de entidades centrais. Outra semelhança é a identificação das bactérias mutantes pela seleção de parentesco. Essa característica também é interessante para as MANETs, pois pode auxiliar na identificação de nós de má-conduta em operações de replicação de dados, por exemplo.

O esquema proposto para a exclusão de nós de má-conduta em sistemas de quórum tem como inspiração esses mecanismos biológicos, em que os nós monitoram a qualidade da interação dos outros nós e os classificam de acordo com o comportamento observado. Assim, os nós são considerados confiáveis quando o comportamento segue o padrão da rede, e são considerados nós de má-conduta quando o comportamento difere do esperado. A partir dessa classificação, os nós buscam selecionar e incluir nos quóruns somente aqueles que se comportam de acordo com o padrão. Desta forma, espera-se diminuir e tolerar a participação de nós de má-conduta nos sistemas de quórum que suportam os serviços de operação em MANETs, amenizando o efeito de tais nós nas operações de um sistema de quórum.

1.3 Contribuições

As contribuições desse trabalho são as seguintes.

- A aferição da eficácia de um sistema de quórum probabilístico, o PAN, diante de ataques de falta de cooperação, temporização e injeção de dados. Essa avaliação permitiu a constatação das vulnerabilidades desses sistemas de quórum, e possibilitou o entendimento das necessidades de segurança em tais sistemas.
- A proposta e a especificação do QS^2 (*quorum system + quorum sensing*), um esquema bio-inspirado para a mitigação de nós de má-conduta dos sistemas de quórum em MANETs. A arquitetura do QS^2 é definida por dois módulos: o módulo de decisão de cooperação e o módulo de monitoramento dos nós. Juntos, eles monitoram e classificam os nós da rede, auxiliando os sistemas de quórum na seleção de nós para participar das operações de replicação.

- A avaliação do QS^2 em um sistema de quórum probabilístico para MANETs. A avaliação mostrou que o QS^2 provê uma melhora considerável na confiabilidade dos dados replicados, com uma boa taxa de detecção. Desta forma, o esquema QS^2 foi capaz de mitigar os nós de má-conduta das operações dos sistemas de quórum, aumentando a confiabilidade dos dados de serviços de operação de rede.
- A análise do uso do QS^2 no apoio aos serviços de operação de rede diante de cenários realísticos de MANETs. O QS^2 foi empregado na mitigação de nós de má-conduta em sistemas de quórum em cenários de ambientes urbanos para a distribuição de informações do comércio local no centro de uma cidade e em ambientes de transporte, na distribuição de informações sobre o tráfego e horários em linhas de ônibus. Os resultados mostraram que o QS^2 é viável nesses cenários, desde que se garanta a entrega das mensagens pelos nós.

1.4 Estrutura da dissertação

Esta dissertação está organizada em sete capítulos. O Capítulo 2 apresenta os fundamentos dos sistemas de quórum clássicos e dos sistemas de quórum para as MANETs. Ele enfatiza o funcionamento do sistema de quórum PAN, específico para MANETs, e relata as vulnerabilidades desse sistema por meio da discussão dos resultados obtidos pela sua avaliação diante de nós de má-conduta.

O Capítulo 3 descreve as características e as deficiências dos sistemas de detecção de nós de má-conduta existentes para MANETs. Também é descrito o funcionamento dos mecanismos de sensoriamento em quórum e de seleção por parentesco, que são mecanismos biológicos empregados pelas bactérias e são a inspiração para o esquema proposto.

O Capítulo 4 apresenta um esquema bio-inspirado para a exclusão de nós de má-conduta nas operações de um sistema de quórum. Esse esquema é denominado QS^2 , e sua arquitetura e detalhes de funcionamento são descritos nesse capítulo.

O Capítulo 5 apresenta uma análise do esquema proposto aplicado a um sistema de quórum diante de nós de má-conduta, considerando métricas de desempenho e eficiência.

O Capítulo 6 avalia o uso do QS^2 no apoio aos serviços de operação de rede em cenários realísticos de MANETs. Os resultados de desempenho e eficiência são apresentados nesse capítulo.

Por fim, o Capítulo 7 descreve as considerações finais e apresenta as atividades futuras.

CAPÍTULO 2

GERÊNCIA DE DADOS DE OPERAÇÃO DE REDE EM MANETS

Este capítulo apresenta os fundamentos desse trabalho. A Seção 2.1 descreve o funcionamento dos sistemas de quórum clássicos, e aborda as características dos diversos tipos de sistemas de quórum existentes, além de descrever em detalhes um sistema de quórum específico para MANETs. A Seção 2.2 apresenta o comportamento que os nós de má-conduta podem apresentar nos sistemas de quórum para MANETs. Por fim, a Seção 2.3 analisa o comportamento do sistema de quórum PAN diante de nós de má-conduta, avaliando o impacto que esses nós podem causar em um sistema de quórum probabilístico para MANETs.

2.1 Sistemas de quórum

Os sistemas de quórum são caracterizados por um conjunto de operações de relacionamento entre dados e um sistema de armazenamento. Essas operações proveem a replicação de dados entre servidores, em uma rede no modelo cliente/servidor. Os sistemas de quórum têm sido utilizados principalmente para a tolerância a falhas de parada (*fail-stop*) em servidores [8]. Além de tolerar esse tipo de falha, eles possibilitam que a replicação seja realizada com um número menor de servidores do que em esquemas de replicação clássicos, como a replicação por máquina de estados [7], que replica os dados em todos os servidores. Logo, menos recursos de comunicação e computação são gastos com o uso dos sistemas de quórum.

Esse modelo de replicação tem sido empregado na resolução de diversos problemas de gerenciamento e controle de aplicações distribuídas. Em redes clássicas, como a Internet, eles toleram falhas em repositórios de dados [8], em serviços de exclusão mútua [25, 26], no consenso [27] e na implementação de memória compartilhada [28, 29, 30]. Já em MANETs, os sistemas de quórum têm sido aplicados em serviços de descoberta de recursos, como DNS e DHCP [9, 31], na economia de energia [10] e no armazenamento distribuído de dados [9].

O princípio do funcionamento dos sistemas de quórum consiste na criação de subconjuntos (quóruns), dentre um universo de n servidores, em que os quóruns devem intersectar entre si. Para a realização da replicação dos dados, os sistemas de quórum proveem operações de escrita e de leitura, sendo que essas operações são executadas nos servidores de um dos quóruns formados, escolhidos aleatoriamente em cada operação de escrita ou de

leitura [8]. A Figura 2.1 ilustra um sistema de quórum \mathcal{Q} , composto por dois quórums, Q_1 e Q_2 , em um conjunto de sete servidores $\{s_1 - s_7\}$. Os servidores s_4 e s_5 fazem parte da intersecção entre os quórums Q_1 e Q_2 e são responsáveis por manter a consistência no sistema. Dessa forma, uma escrita realizada no quórum Q_1 é observada em uma posterior leitura no quórum Q_2 por meio dos nós s_4 e s_5 , que participam de ambas as operações.

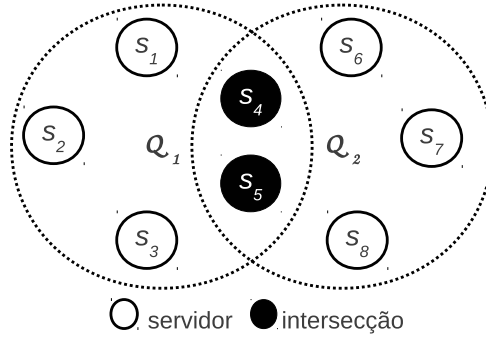


Figura 2.1: Construção de um sistema de quórum \mathcal{Q}

Os sistemas de quórum empregam dois protocolos principais: um protocolo de escrita e um protocolo de leitura. Esse conjunto de protocolos caracteriza a **estratégia de acesso** aos quórums [32]. Ela determina a forma como a replicação é realizada no sistema. Diante da possibilidade de criar estratégias de acesso diferentes, diversos sistemas de quórum foram propostos para prover diferentes níveis de consistência aos dados, tais como a consistência global [9] e a baseada em tempo [33].

Nesse trabalho, os sistemas de quórum existentes foram classificados em dois grandes grupos: os sistemas de quórum bizantinos e os sistemas de quórum probabilísticos, ilustrados na Figura 2.2. O grupo dos sistemas de quórum bizantinos engloba os sistemas de quórum tradicionais, que necessitam de canais confiáveis e comunicação síncrona, e portanto, são mais adequados para redes fixas que empregam um modelo cliente/servidor. O grupo dos sistemas de quórum probabilísticos é composto pelos sistemas de quórum que não dependem de canais confiáveis ou sincronismo, o que os tornam adequados para redes dinâmicas como as MANETs. Os sistemas de quórum bizantinos são propostos principalmente para serviços que envolvem a validação da operação dos servidores, como exclusão mútua e memória compartilhada, enquanto que os sistemas de quórum probabilísticos focam na distribuição e no armazenamento de dados dos serviços de operação de rede, como os serviços de localização de recursos. Nas seções seguintes, os sistemas de quórum bizantinos e os sistemas de quórum probabilísticos são explicados com maiores detalhes.

2.1.1 Quórums bizantinos

O grupo dos sistemas de quórum bizantinos se caracteriza pela tolerância a falhas bizantinas, ou seja, falhas em que os nós se comportam arbitrariamente [8]. A estratégia de

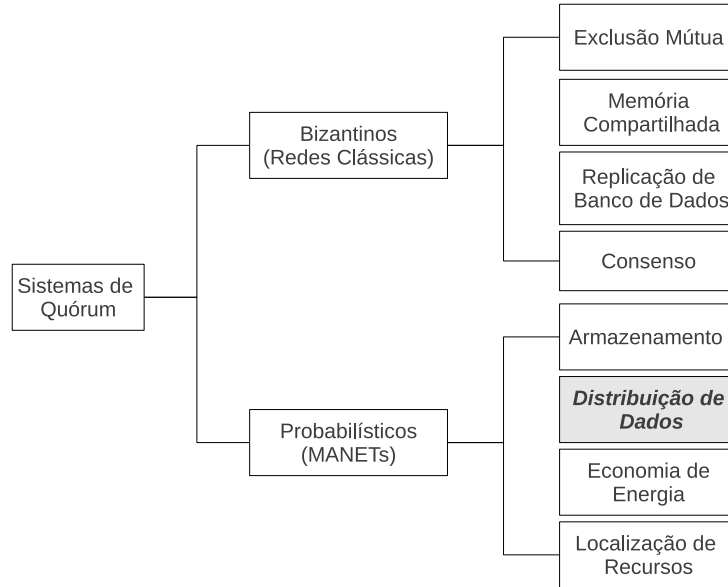


Figura 2.2: Aplicação dos sistemas de quórum em redes e serviços

acesso empregada nesses sistemas foca no reforço da intersecção entre os quóruns, com o objetivo de reduzir a probabilidade de falhas simultâneas entre os servidores que fazem a intersecção. Comumente emprega-se $2f + 1$ [8] ou $3f + 1$ [34] servidores na intersecção, em que f é o número máximo de falhas concorrentes toleradas pelo sistema.

Uma crítica geralmente imposta a esse sistema é o número estático de servidores na intersecção, não permitindo uma reconfiguração do sistema de acordo com o número de faltas presentes [29]. Apesar de existirem sistemas de quórum dinâmicos [35], eles não são adequados para as MANETs devido aos protocolos utilizados para a reconfiguração dos quóruns. Além disso, na maioria dos casos, a quantidade de falhas que o sistema tolera é um número pessimista, ou seja, f é configurado para tolerar um grande número de falhas que pode não corresponder com o ambiente, implicando em mais recursos utilizados na replicação. Além do mais, as condições que esses sistemas de quórum impõem para que a replicação seja efetivamente tolerante a falhas bizantinas são difíceis de garantir, tais como a existência de comunicação síncrona e de canais sem perdas de mensagens [8]. Eles também estabelecem critérios rigorosos para a construção das intersecções, dificultando sua implementação em MANETs, principalmente devido à mobilidade e à entrada e saída de nós da rede. Essas características prejudicam a construção e a manutenção das intersecções nesses tipos de sistemas de quórum.

2.1.2 Quóruns probabilísticos

Os sistemas de quórum probabilísticos foram propostos para a adaptação dos sistemas de quórum tradicionais, de forma que possam ser empregados em sistemas dinâmicos [11]. Para isso, eles relaxam as propriedades da intersecção entre os quóruns de modo que

elas aconteçam com uma determinada probabilidade. Dessa forma, as intersecções são probabilísticas, ao contrário dos sistemas de quórum bizantinos, em que as intersecções são estáticas e restritas.

As estratégias de acesso para esses sistemas de quórum precisam garantir a intersecção entre os quórums com uma alta probabilidade [36]. Além disso, os servidores que compõem a intersecção não são estáticos e são escolhidos probabilisticamente, tornando esse tipo de sistema de quórum viável para o uso em sistemas dinâmicos. Por esses motivos, eles têm sido empregados principalmente em MANETs, auxiliando serviços como localização de recursos [9, 31], economia de energia [10], armazenamento distribuído [9] e serviços de operação da rede [37].

Enquanto os sistemas de quórum bizantinos proveem resiliência para a replicação em ambientes bizantinos em troca de um maior uso de computação e comunicação, os sistemas de quórum probabilísticos relaxam essas restrições para se adaptar aos sistemas dinâmicos. Contudo, eles são capazes de proporcionar somente uma consistência mais fraca. Por isso, esse tipo de sistema de quórum é mais indicado para aplicações cuja garantia da disponibilidade é mais relevante que o custo de lidar com eventuais inconsistências [11]. O ambiente das MANETs justifica essa troca, pois os sistemas de quórum probabilísticos proporcionam uma melhor adaptação às constantes mudanças de sua topologia. Além disso, diferentes estratégias de acesso podem ser empregadas, ajustando o nível de consistência de acordo com o serviço fornecido [38].

Diversos sistemas de quórum probabilísticos foram propostos para MANETs, como o PAN (*Probabilistic quorum system for Ad hoc Networks*) [9], o MDQ (*Mobile Dissemination Quorum system*) [10] e o Timed [33]. Esses sistemas têm como característica o uso da probabilidade para a escolha de participantes para os quórums, e consequentemente, possuem o mesmo comportamento. Contudo, o PAN propõe o uso de um número reduzido de mensagens para a replicação ao introduzir o conceito de quórums assimétricos em MANETs. Dessa forma, ele foi escolhido neste trabalho para representar a classe de sistemas de quórum probabilísticos para MANETs. A próxima seção explica os detalhes do funcionamento das operações do PAN.

2.1.3 Quórums probabilísticos para redes *ad hoc* (PAN)

O PAN (*Probabilistic quorum system for Ad hoc Networks*) [9] é um dos primeiros sistemas de quórum probabilísticos criados para MANETs, e tem como objetivo tornar pequenos dados altamente disponíveis na rede por meio da replicação. Tendo como base os sistemas de quórum probabilísticos tradicionais [11], ele emprega uma estratégia de acesso focada na distribuição dos dados de uma forma mais relaxada, de modo a considerar as características das MANETs. O PAN também é o pioneiro no uso de quórums assimétricos [34], isto é, quórums de tamanho diferente, e na aplicação de estratégias de acesso diferentes

para as leituras e as escritas.

O PAN classifica os nós da rede em três entidades: os servidores, os clientes e os agentes. Os servidores são entidades que replicam os dados entre si e são organizados em um conjunto chamado StS (*Storage Set*). Os clientes são entidades que emitem requisições de leitura e de escrita para os servidores do StS. O agente é a denominação de um servidor do StS que está atendendo a uma requisição de um cliente, visto que o cliente escolhe aleatoriamente um servidor para fazer a mediação entre ele e o StS.

A estratégia de escrita dos dados empregada pelo PAN tem como base o comportamento de protocolos epidêmicos. Os protocolos epidêmicos são caracterizados pelo comportamento de propagação de forma eficiente e escalar, em que cada entidade propaga um dado para outras. Isso pode acontecer periodicamente ou sob demanda [39]. Já a leitura é realizada por mensagens *unicast*. Os dados replicados no StS são localizados na rede por meio de um identificador único em conjunto com um carimbo de tempo (*timestamp*) associado a ele. Um valor de *timestamp* maior significa que o dado é mais recente. O comportamento epidêmico do protocolo de escrita utilizado para replicar os dados no StS ocorre de forma que os nós atualizam os dados a medida em que recebem as escritas. Além disso, os nós devem repassar esse dado para os outros nós da rede, dando continuidade à replicação, até que todos os nós do StS tenham o dado inicialmente replicado. Na operação de leitura de um dado no sistema, o agente consulta um determinado número de servidores do StS, pesquisando pelos valores do dado que esses servidores possuem. Ao receber as respostas, o agente verifica o valor mais atual e envia a resposta para o cliente.

Ao receber uma requisição de escrita de um cliente, o agente atualiza o dado contido em seu repositório e repassa essa escrita para outros F nós do StS. O parâmetro F (*fanout*) determina o número de nós para os quais deve-se repassar o dado, e no PAN ele é definido de forma que os quóruns de escrita se intersectem com alta probabilidade com os quóruns de leitura. No PAN, a propagação dos dados acontece a cada T intervalo de tempo, em que o valor de T é igual para todos os nós do sistema. Os nós que recebem um dado enviado pelo cliente formam o quórum de escrita, chamado de Q_w .

A execução de uma escrita no PAN é exemplificada na Figura 2.3. Nela, o StS é composto por sete servidores $\{s_0 - s_6\}$ e $F = 2$. O cliente envia uma requisição de escrita do dado v para o servidor s_0 , que passa a atuar como o agente dessa operação. O agente atualiza o dado v em seu repositório com o valor enviado pelo cliente e então propaga o novo valor de v para outros F servidores, nesse caso, s_1 e s_2 . Esses servidores seguem o protocolo e enviam o novo valor do dado v para mais F servidores cada um. Isso se repete até que todos os servidores tenham seus repositórios atualizados, formando o quórum de escrita Q_w . Um nó que recebe mais de uma vez a mesma atualização não a propaga novamente.

O número de nós consultados pelo agente em uma operação de leitura é pré-definido pelo sistema, desta forma o quórum de leitura, chamado Q_r , sempre possui o mesmo

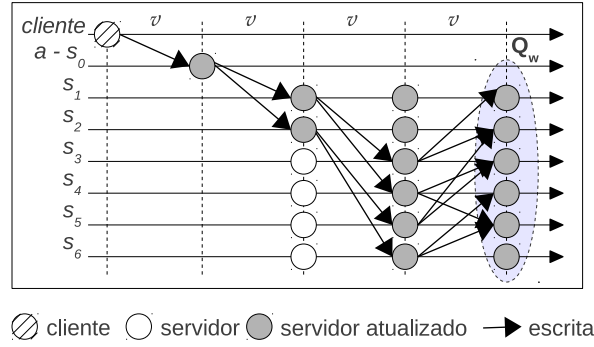


Figura 2.3: Operação de escrita no PAN

número de participantes. A quantidade de nós no quórum de leitura é calculado para que a intersecção com os quóruns de escrita se dê com uma alta probabilidade. Ao receber uma requisição de leitura, o agente consulta os nós do quórum de leitura, e aguarda pela resposta desses servidores. Essa etapa tem um tempo de espera (*timeout*), caso os servidores do quórum não respondam. Isso pode acontecer devido ao atraso da rede ou pela perda de pacotes. Ao expirar o *timeout*, o agente responde ao cliente com o dado mais atual entre aqueles que foram recebidos. Um agente que recebe um dado mais recente do que o seu, atualiza o respectivo dado em seu repositório e dá início à propagação desse dado. Nesse caso específico, a leitura é também uma escrita.

A Figura 2.4 ilustra o funcionamento de uma leitura no PAN, com um quórum de leitura formado por $Q_r = 4$ servidores, e o StS é composto por sete servidores $\{s_0 - s_6\}$. O cliente emite uma requisição de leitura para o servidor s_0 , agora o agente dessa operação de leitura. O servidor s_0 consulta outros três servidores para compor o quórum de leitura, nesse caso os servidores s_1 , s_2 e s_3 . Esses servidores respondem ao agente se possuem um dado mais atual que o dado enviado pelo agente para consulta. O agente aguarda por um tempo pelas respostas dos servidores do quórum e responde ao cliente com o dado mais atual dentre as respostas recebidas.

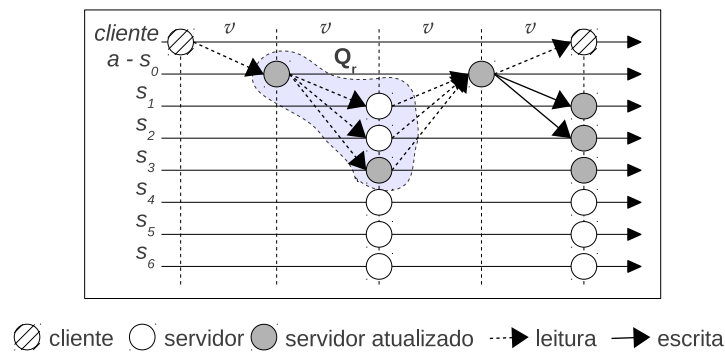


Figura 2.4: Operação de leitura no PAN

As estratégias de leitura e de escrita foram projetadas para lidar com a mobilidade, a

partição da rede e os recursos escassos das MANETs. Entretanto, o PAN foca no desempenho e na disponibilidade dos dados, e não tem como objetivo fornecer mecanismos de defesa contra nós de má-conduta. Dessa forma, nós egoístas e maliciosos podem participar de uma MANET e prejudicar o funcionamento do sistema de quórum. Além disso, o PAN está sujeito a ataques empregados diretamente em sistemas de replicação de dados, como os ataques de temporização e os de injeção de dados [14]. A próxima seção define um conjunto de ataques que podem prejudicar os sistemas de quórum nas MANETs.

2.2 Tipos de ataques no PAN

O sucesso das estratégias de acesso empregadas pelo PAN depende da colaboração entre os nós da rede, tanto no progresso das operações de leitura quanto das operações de escrita. Nós de má-conduta que venham a intervir no funcionamento dos protocolos de escrita e de leitura podem comprometer a execução dessas operações. Esses nós de má-conduta podem se manifestar de duas formas: como nós egoístas ou como nós maliciosos. Uma classificação mais detalhada quanto aos tipos de nós de má-conduta é encontrada em [40]. Os nós egoístas não colaboram com as operações da rede, visando economia de recursos [41]. Já os nós maliciosos podem interceptar e modificar mensagens, ou ainda injetar dados maliciosos na rede com o intuito de degradar o sistema [42]. Além desses ataques comuns em MANETs, o PAN e outros sistemas de replicação também podem sofrer ataques de injeção de dados e de atraso na replicação dos dados [14], que são ataques específicos dos sistemas de replicação de dados.

A presença de nós de má-conduta na rede pode prejudicar a replicação de várias formas. Nas operações de escrita, os nós egoístas podem impedir o progresso da propagação das escritas e os nós maliciosos podem injetar dados inconsistentes durante a propagação dos dados. Nas operações de leituras, os servidores egoístas podem não responder às requisições emitidas pelos agentes, forçando-os a enviar sua própria resposta para os clientes. Além disso, os servidores maliciosos podem modificar o valor enviado como resposta ao cliente. Nesse caso, os agentes confiam no dado enviado pelos servidores maliciosos e também atualizam o seu repositório com o valor falso, ocasionando o início da propagação desse valor para todo o StS. A princípio os clientes bizantinos não foram considerados, e os servidores de má-conduta são intrusos que possuem permissão para participar da rede. A descrição do comportamento desses nós é apresentada a seguir.

2.2.1 Falta de cooperação

Os ataques de *falta de cooperação* podem ocorrer em ambas as operações implementadas pelo PAN. Eles acontecem quando os nós de má-conduta agem de forma egoísta, e se recusam a desempenhar sua parte no protocolo de replicação [41]. A Figura 2.5(a) ilustra

o comportamento de um agente egoísta em uma operação de leitura. Um cliente emite uma requisição de leitura de um dado v para o servidor s_0 , que é um nó egoísta. O agente egoísta s_0 não consulta o quórum de leitura, aguarda o *timeout* da operação e responde ao cliente com o dado que ele possui, que pode estar desatualizado. Esse tipo de ataque é difícil de detectar, pois os clientes recebem as respostas dos agentes independente da consulta ao quórum de leitura.

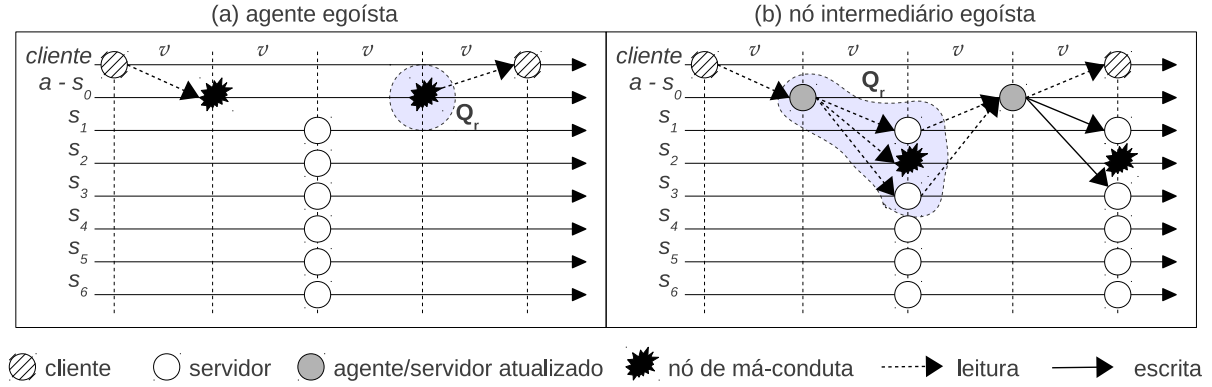


Figura 2.5: Ataque de falta de cooperação na leitura

Ainda nas operações de leitura, um nó egoísta que faça parte do quórum de leitura simplesmente não responde às requisições dos agentes. A Figura 2.5(b) ilustra essa situação, onde o cliente emite um pedido de leitura de um dado v para o agente, que consulta o quórum de leitura composto por s_1, s_2 e s_3 . O servidor s_2 é egoísta, e portanto, ignora a requisição do agente. Diante dessa situação, a operação de leitura ainda pode terminar corretamente com a resposta dos demais componentes do quórum, ou se o próprio agente possuir o dado atualizado.

Nas operações de escrita, os nós egoístas não atualizam seus repositórios e nem propagam novos dados para os outros nós do StS. A Figura 2.6(a) ilustra o comportamento de um agente egoísta agindo nas operações de escrita. O cliente emite uma requisição de escrita do dado v para o servidor s_0 , que é um nó egoísta. Como o nó s_0 não propaga a escrita, a operação não é finalizada. Isso também ocorre com um servidor egoísta, que não escreve o novo dado no seu repositório nem propaga para os demais nós os novos dados que ele recebe dos agentes. Como resultado, a propagação das escritas não progride.

A Figura 2.6(b) ilustra o caso de uma escrita, com um nó egoísta participando do quórum de escrita. O cliente emite uma requisição de escrita para o servidor s_0 , que segue o protocolo de escrita e a propaga para outros F servidores. O servidor s_2 é um nó egoísta e não dá continuidade a essa escrita. Dessa forma, a propagação da escrita depende dos demais servidores no quórum. Esse ataque pode ter um impacto maior se o quórum de escrita possuir mais nós egoístas na mesma operação de escrita.

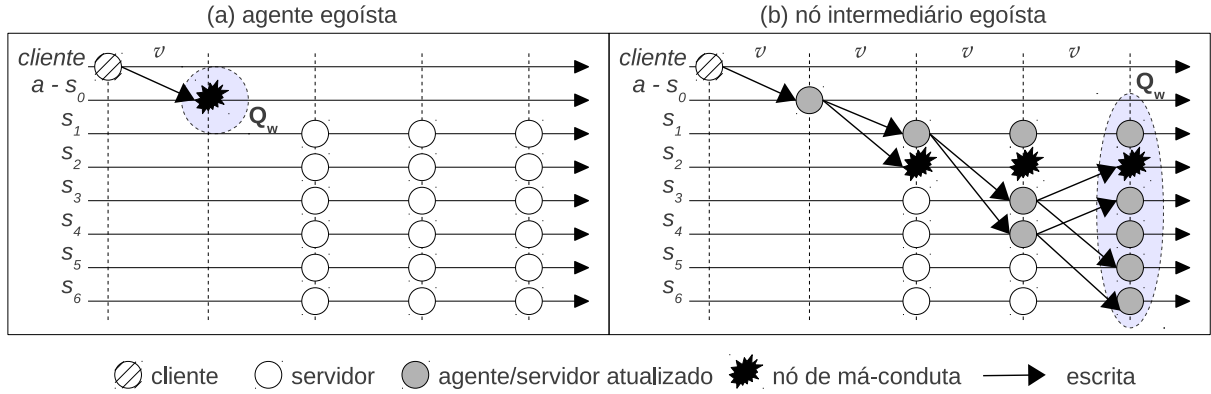


Figura 2.6: Ataque de falta de cooperação na escrita

2.2.2 Temporização

O intervalo de tempo entre as propagações de uma nova escrita é um parâmetro importante para o PAN. Ele determina a velocidade da propagação de uma escrita, e quanto maior a velocidade, mais rápida é a formação do quórum de escrita. O ataque em que os nós atrasam deliberadamente a propagação da escrita é chamado de *temporização* [14], e é causado por nós maliciosos.

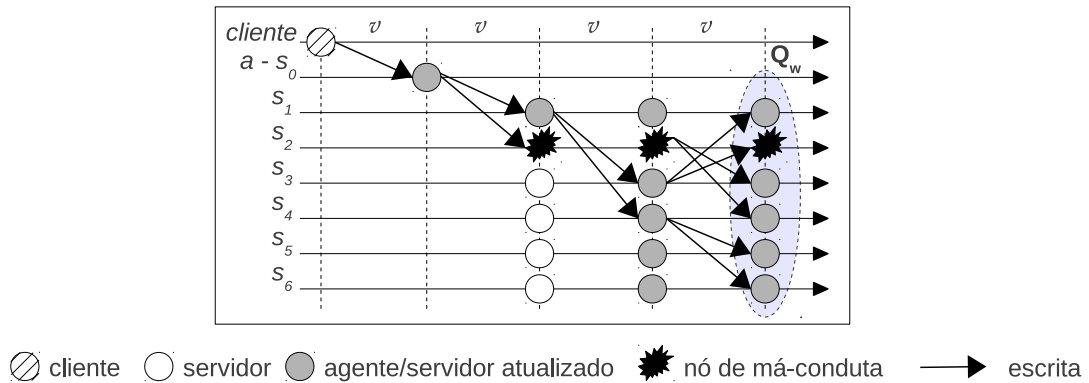


Figura 2.7: Ataque de temporização

Nesse ataque, o intervalo de tempo T estabelecido pelo sistema é ignorado pelo nó malicioso, que determina arbitrariamente seu próprio intervalo T . A Figura 2.7 apresenta o caso de um ataque de temporização no PAN. Após o cliente enviar uma requisição de escrita para o servidor s_0 , o servidor começa a propagação do dado para o restante dos nós no StS. O servidor s_2 é um nó malicioso, e arbitrariamente atrasa a propagação da escrita. Com esse comportamento, o sistema demora mais para realizar a escrita em todos os nós do StS.

2.2.3 Injeção de dados

O ataque de *injeção de dados* consiste em nós que recebem os dados, tanto de clientes quanto de outros servidores, e os modificam para um valor arbitrário [43]. Esse ataque pode acontecer nas operações de leitura e de escrita. O resultado pode ser um sistema com dados inconsistentes, que não é capaz de auxiliar uma aplicação. A Figura 2.8(a) mostra o comportamento de um agente malicioso na operação de leitura. Ao receber a requisição de leitura do cliente, o agente s_0 fabrica um dado falso e substitui o pacote do dado enviado. Ele também identifica o dado falso como mais atual, e então consulta o quórum de leitura. Ao receber a requisição do agente, os servidores identificam o valor enviado por ele como mais recente para aquele dado. Isso força os servidores do quórum de leitura a iniciar a propagação desse dado para o restante do StS. Além disso, como este valor é mais recente, o quórum de leitura não responde ao agente, e dessa forma, o agente envia como resposta para o cliente um valor fabricado por ele.

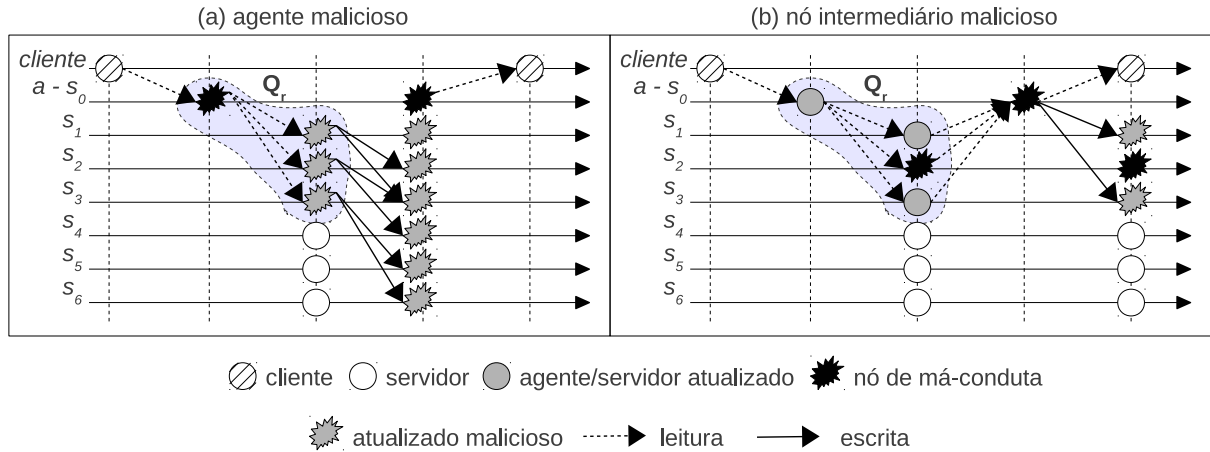


Figura 2.8: Ataque de injeção de dados na leitura

Quando o nó malicioso participa do quórum de leitura, o impacto desse ataque é menor, como mostra a Figura 2.8(b). O cliente envia uma requisição de leitura de um dado v para o servidor s_0 , que consulta o quórum de leitura. O servidor s_2 é um nó malicioso e responde ao agente com um dado falso. Ao receber as respostas dos nós do quórum de leitura, a resposta do nó malicioso será mais recente, identificada pelo seu *timestamp* falso. O agente atualiza seu repositório e começa a propagação desse dado, acreditando que ocorreu uma nova escrita no sistema. Além disso, o resultado enviado ao cliente também será o valor falso.

Já nas operações de escrita, os nós maliciosos podem representar uma grande ameaça ao sistema de replicação. A Figura 2.9(a) ilustra o caso em que o agente é malicioso. Semelhante ao ataque de injeção de dados nas leituras, um agente malicioso danifica o sistema rapidamente, uma vez que qualquer requisição enviada a ele resulta em uma escrita incorreta. Quando um cliente envia uma requisição de escrita de um dado v

para o servidor s_0 , o agente modifica o valor e o *timestamp* a ser escrito, dando início à propagação de um dado falso. Nesse caso, todos os servidores do quórum de escrita atualizam seu repositório com o dado falso. Um nó malicioso em um quórum de escrita, como mostra a Figura 2.9(b), modifica o dado recebido de outros servidores sempre que for contatado para colaborar com a propagação.

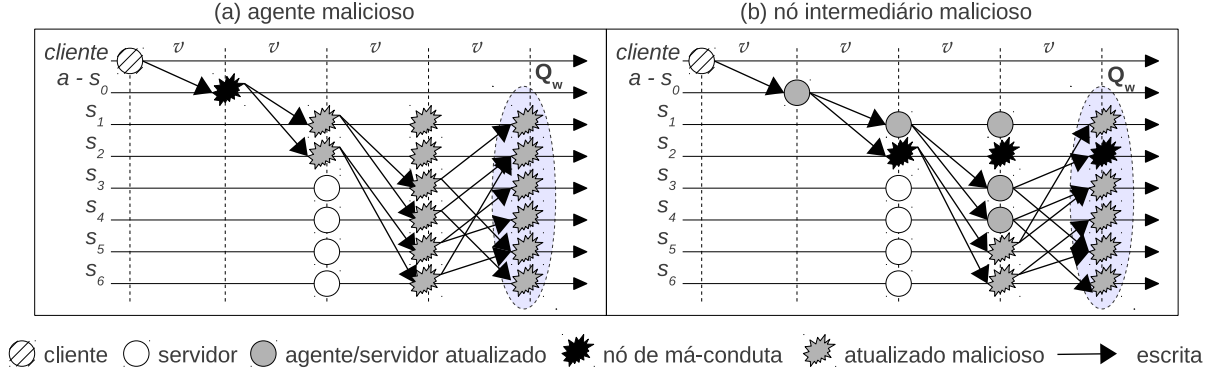


Figura 2.9: Ataque de injeção de dados na escrita

A próxima seção descreve uma avaliação do PAN diante desses três tipos de ataques, mostrando o comportamento de cada ataque e seu impacto nas vulnerabilidades existentes no PAN.

2.3 Avaliação do PAN diante de má-conduta

O PAN foi implementado no simulador *Network Simulator* (NS-2) versão 2.33, e os parâmetros de simulação são baseados em [9]. Para a simulação de cenários com nós de má-conduta, foram implementados e adicionados ao PAN nós egoístas e maliciosos, que se comportam conforme o descrito na Seção 2.2. Informações complementares sobre os parâmetros utilizados e os testes realizados são encontradas em [44, 16].

Nos cenários simulados, a rede é composta por 50 nós, sendo que 25 são servidores do StS e são escolhidos aleatoriamente no início da simulação. Os nós comunicam-se por meio de um canal sem fio, seguindo o modelo de propagação *TwoRayGround* [45], e a movimentação segue o padrão *Random Waypoint* [46]. Os nós se movimentam em uma área de 1000x1000 metros, e a velocidade máxima dos nós varia de 2m/s, 5m/s, 10m/s e 20m/s com os respectivos tempos de pausa de 10s, 20s, 40s e 80s. As velocidades consideradas são similares à velocidade de movimentação de pedestres e ciclistas. O raio de alcance da antena de todos os nós é de 250 metros, e o protocolo de roteamento utilizado é o AODV (*Ad Hoc On-Demand Distance Vector*) [47].

As escritas são propagadas a cada $T = 200\text{ms}$, e o *fanout* é igual a dois servidores. Os quóruns de leitura (Q_r) são compostos por quatro servidores, incluindo o agente e o quórum de escrita (Q_w) é composto por todos os nós que recebem um dado novo. Os

pacotes com as mensagens de leitura e de escrita têm 128 *bytes*, suficiente para o tipo de dado de controle que pretende-se replicar. Seguindo o proposto em [9], cada nó define o intervalo de envio de leituras e de escritas por uma distribuição de *Poisson*, em que as escritas acontecem em um intervalo médio de 100 segundos, e as leituras têm um intervalo médio de aproximadamente 36 segundos.

A porcentagem de nós de má-conduta (f) varia em 20%, 28% e 36% dos servidores do StS, o que corresponde a 5, 7 e 9 nós de má-conduta, respectivamente. Um número maior de nós de má-conduta do que o considerado nas simulações resulta em quóruns inteiramente comprometidos, em que nenhuma leitura correta é concluída diante de ataques de injeção de dados. Dessa forma, o custo de lidar com todas as inconsistências geradas pelos nós de má-conduta é maior do que o benefício para a rede. Os nós de má-conduta agem de forma egoísta ou maliciosa em todo o período da simulação, e não mudam a forma do seu comportamento, ou seja, um nó egoísta não se comporta de forma maliciosa e da mesma forma, um nó malicioso não se comporta de forma egoísta. A Tabela 2.1 resume os principais parâmetros aplicados nas simulações. Os resultados apresentados são a média de 35 simulações com um intervalo de confiança de 95%, sendo que o tempo de vida da rede em cada simulação é de 1500 segundos. A comparação entre os resultados do PAN com e sem ataques é realizada através de pontos percentuais.

Tabela 2.1: Principais parâmetros de simulação dos cenários de avaliação

Parâmetro	Valor
Quantidade de nós	50
Quantidade de nós no StS	25
Tempo de vida da rede	1500s
Área de movimentação	1000x1000 metros
Velocidades máximas	2m/s, 5m/s, 10m/s 20m/s
Tempo de pausa	10s, 20s, 40s, 80s
Raio de transmissão	250 metros
$Fanout(F)$	2
Quórum de leitura (Q_r)	4 servidores
Intervalo de propagação	200ms, 400ms, 800ms, 3000ms
Quantidade de nós de má-conduta	20%, 28% e 36%
Intervalo médio entre as escritas	100s
Intervalo médio entre as leituras	36s

Duas métricas foram empregadas na avaliação do PAN: o *grau de confiabilidade* (G_c) [9] e a *quantidade de nós de má-conduta* (Q_m) nas operações de leitura. O G_c quantifica a probabilidade de intersecção entre os quóruns de leitura e de escrita, representando também a quantidade de leituras corretas obtidas pelos clientes. Consideram-se corretas as leituras que retornam o valor de uma escrita previamente realizada no sistema, assim como o de uma escrita em progresso no momento da leitura [9]. A métrica G_c é definida na Equação (2.1), em que C_r representa as leituras que obtiveram resultados atualizados e $|R|$ a quantidade de requisições de leituras emitidas pelos clientes.

$$G_c = \frac{\sum C_r}{|R|} \quad (2.1)$$

A métrica Q_m representa o número de vezes que um nó de má-conduta participou de operações de leitura no PAN. Essa métrica é contabilizada para os ataques de falta de cooperação e injeção de dados, pois o ataque de temporização acontece somente nas escritas. O Q_m é definido pela Equação (2.2), onde Mr_i representa uma operação de leitura em que pelo menos um membro do quórum de leitura é de má-conduta.

$$Q_m = \frac{\sum Mr_i}{|R|} \forall i \in R, \quad \text{onde} \quad Mr_i = \begin{cases} 1 & \text{se } \exists f \in Q_r(R_i) \\ 0 & \text{caso contrário} \end{cases} \quad (2.2)$$

2.3.1 Ataque de falta de cooperação

A Figura 2.10 ilustra os resultados obtidos com a execução do PAN diante de ataques de falta de cooperação. Os resultados estão agrupados por velocidades, e apresentados para cada quantidade de nós de má-conduta considerada nas simulações. Os resultados apresentados são comparados aos resultados obtidos pelo PAN sem a presença de nós de má-conduta, que apresenta um G_c de aproximadamente 99%. Os resultados do PAN sem ataques são detalhados em [9].

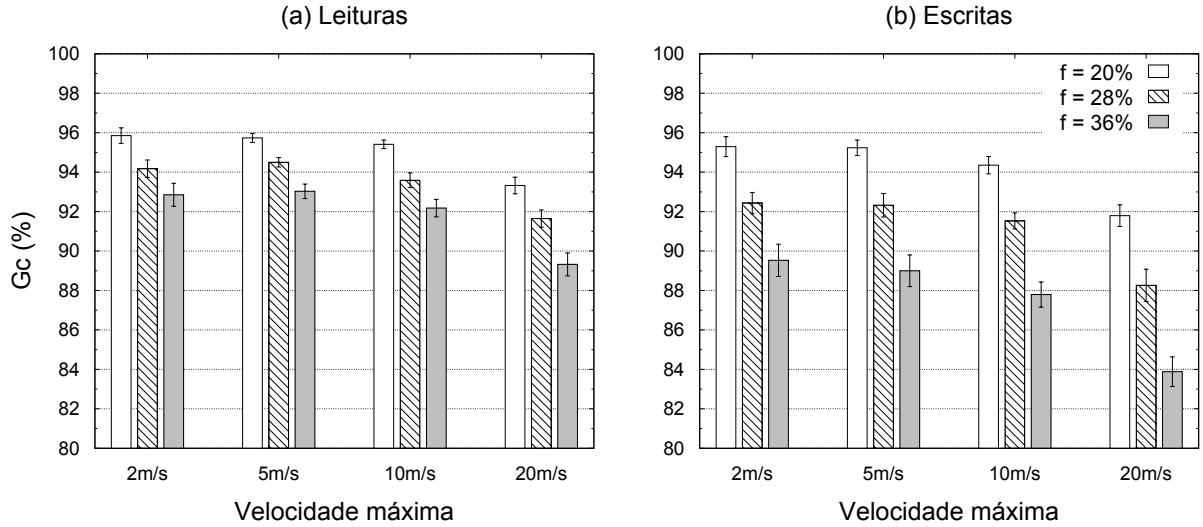


Figura 2.10: G_c com ataque de falta de cooperação

Primeiramente, observa-se que em ambas as operações, o G_c diminui com o aumento da porcentagem de nós egoístas. Além disso, as operações de escrita são mais afetadas do que as operações de leitura, independente do número de nós egoístas e da velocidade dos nós. Também nota-se que maiores velocidades resultam em um menor G_c . Isso porque os nós encontram dificuldade na entrega de pacotes devido às quebras de enlace e manutenção de rotas para os nós. Nas operações de leitura, em um cenário com 20% de nós egoístas

no StS e nós movimentando-se a uma velocidade de 20m/s, o resultado para o G_c foi de 93,3%, como apresenta a Figura 2.10(a). Já nas operações de escrita, um cenário com 20% de nós egoístas resultou em um G_c igual a 91,7% com nós movimentando-se a 20m/s, conforme a Figura 2.10(b). Um comportamento similar acontece com o sistema com 28% e 36% de nós egoístas.

Observa-se também que o G_c tem seu pior desempenho em operações de escrita com 36% de nós egoístas, com uma velocidade máxima de 20m/s. O G_c nesse caso é de 89,3%. Isso ocorre porque nessa situação, praticamente a metade do StS está comprometida, impossibilitando o sistema de concluir a replicação de forma correta.

2.3.2 Ataque de temporização

O resultado obtido para o G_c do PAN diante de ataques de temporização é ilustrado na Figura 2.11. Nesse ataque, foram avaliados cenários em que os nós maliciosos atrasam a propagação das escritas em 400ms, 800ms e 3000ms, ao invés dos 200ms definidos pelo sistema. Observa-se que o G_c diminui proporcionalmente com o aumento do atraso na propagação. Para todas as velocidades, esse comportamento se repete. Além disso, o G_c é menor conforme a porcentagem de nós maliciosos aumenta.

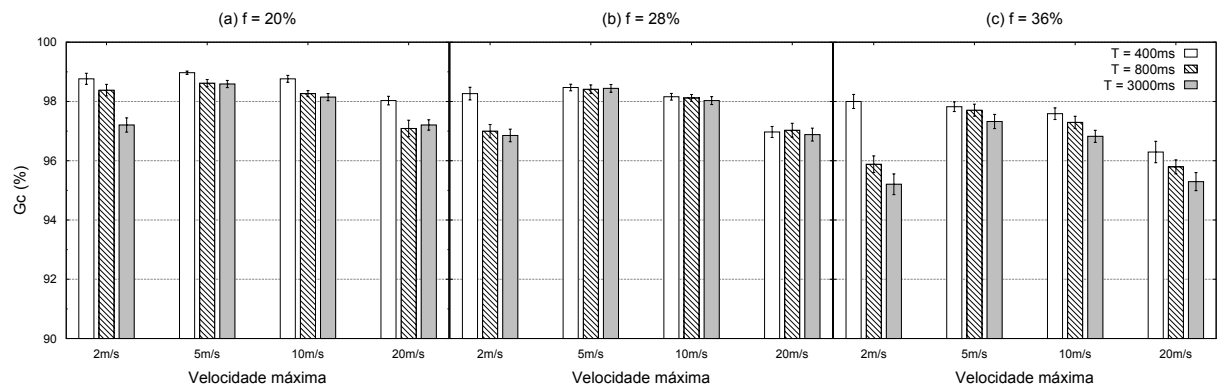


Figura 2.11: G_c com ataque de temporização

Nota-se também que o G_c é mais afetado em maiores velocidades. Em um cenário com 20% de nós maliciosos e atraso na propagação de 400ms, os nós movimentando-se a 5m/s obtêm um G_c de 98,9%, enquanto que o mesmo cenário a uma velocidade de 20m/s o G_c é de 98,2%. Ainda sim, percebe-se que o impacto do ataque de temporização é menor do que o do ataque de falta de cooperação. Isso acontece porque os nós que não são maliciosos conseguem distribuir os dados de forma eficaz, fazendo com que os nós maliciosos que atrasam a propagação não influenciem na entrega dos dados no StS.

2.3.3 Ataque de injeção de dados

O ataque de injeção de dados afeta significativamente o desempenho do PAN. A forma como as operações do PAN são realizadas facilitam a interação de nós maliciosos durante a execução dos protocolos de leitura e escrita, e eventualmente os nós maliciosos são contatados para ajudar na execução do protocolo. A Figura 2.12(a) mostra os resultados obtidos pela simulação quando o PAN enfrenta ataques do tipo de injeção de dados nas leituras. Observa-se que para todos os cenários simulados, o G_c é menor que 50%. Conforme a porcentagem de nós maliciosos aumenta, a confiabilidade no sistema diminui drasticamente. Com 20% de nós maliciosos movimentando-se a 2m/s, o G_c é de 31,1%. Aumentando o número de nós maliciosos para 28%, o G_c cai para 24%.

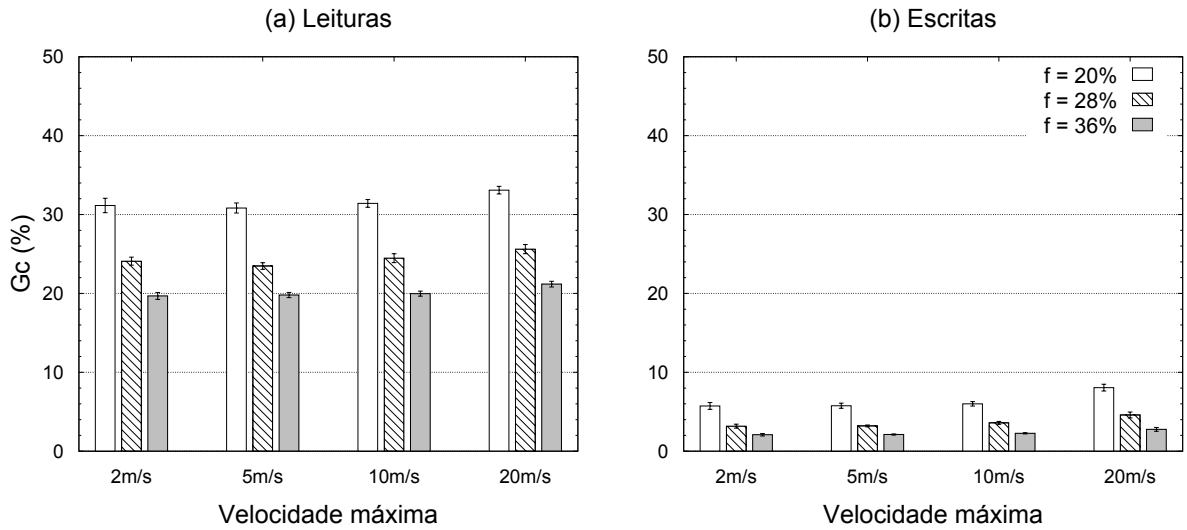


Figura 2.12: G_c com ataque de injeção de dados

Percebe-se também que o ataque de injeção de dados prejudica as operações de escrita de um modo mais acentuado em relação aos prejuízos causados pelos ataques de falta de cooperação e temporização na escrita. Nesse caso, o G_c é menor que 10% em todos os cenários, como mostra a Figura 2.12(b). Por exemplo, nós movimentando-se a 2m/s em cenários com 20% de nós maliciosos, o G_c é de somente 5,7%. Com 28% de nós maliciosos no StS e a mesma velocidade, o G_c cai para 3,1%. Entretanto, esse ataque apresenta um comportamento atípico em relação aos outros dois ataques: conforme a velocidade dos nós aumenta, o G_c também aumenta. Em cenários com 20% de nós maliciosos e velocidade de 10m/s, o G_c é de 6%, e aumentando a velocidade para 20m/s, o G_c tem um aumento de 2%. Isso acontece pelas próprias características das MANETs, em que maiores velocidades dificultam a manutenção de rotas atualizadas para os destinos, e consequentemente, menos pacotes são entregues. Isso implica na perda da efetividade de um nó malicioso ao entregar seus dados falsos pela rede.

2.3.4 Participação de nós de má-conduta em quóruns de leitura

Para os ataques que ocorrem nas operações de leituras, foram contabilizados o número de leituras afetadas por nós de má-conduta, ou seja, leituras em que ao menos um nó de má-conduta participou do quórum de leitura. A Figura 2.13(a) mostra os resultados obtidos no ataque de falta de cooperação. Em tal ataque, como esperado, conforme o número de nós egoístas aumenta, a quantidade de quóruns de leitura afetados por pelo menos um nó egoísta também aumenta. Com 20% de nós egoístas presentes no StS e velocidade de 2m/s, 41,5% dos quóruns de leitura formados tiveram participação de nós egoístas. Aumentando o número de nós egoístas para 36%, 57,9% dos quóruns de leitura são afetados. Esse comportamento é esperado, já que a probabilidade de escolher um nó de má-conduta aumenta com a presença de mais nós de má-conduta na rede.

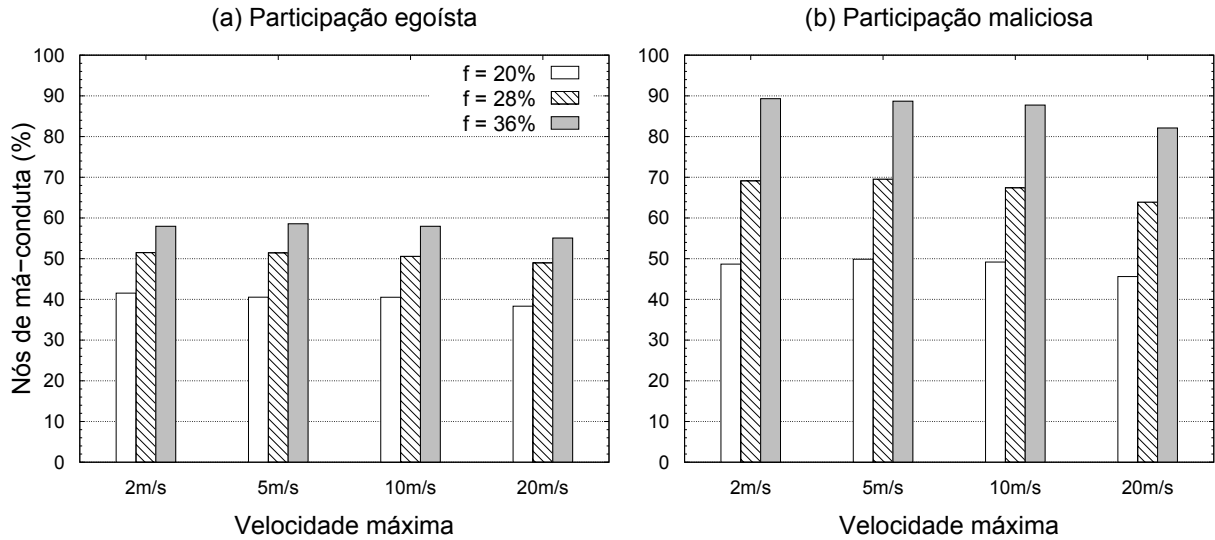


Figura 2.13: Q_r afetados por nós egoístas e maliciosos

No ataque de injeção de dados, apresentado na Figura 2.13(b), a quantidade de leituras afetadas por nós maliciosos aumenta com o aumento do número de nós maliciosos. Com 20% de nós maliciosos e nós movimentando-se a uma velocidade de 2m/s, 45,6% dos quóruns de leitura são comprometidos, enquanto que com 36% de nós maliciosos, 89% dos quóruns são comprometidos. A velocidade máxima dos nós interfere no número de quóruns de leitura comprometidos. Conforme a velocidade aumenta, a participação de nós maliciosos nos quóruns de leitura diminui. Isso ocorre pelos mesmos motivos explicados anteriormente, em que a velocidade máxima dos nós influencia na entrega dos pacotes.

A Tabela 2.2 sintetiza o impacto dos nós de má-conduta nas operações de leitura e de escrita do PAN, tendo como parâmetro o G_c do PAN sem ataques, que é de aproximadamente 99%. No ataque de falta de cooperação na operação de leitura, os nós de má-conduta degradam a confiabilidade do PAN em aproximadamente 5%. Já nos ataques na operação de escrita, o impacto na confiabilidade é de 10%. Nos ataques de temporiza-

ção, a perda de confiabilidade é de aproximadamente 3%, o que caracteriza a menor das ameaças para o PAN. Já os ataques de injeção de dados apresentam um grande impacto na confiabilidade do PAN, sendo que nas operações de leitura o impacto é de 73% e nos ataques de injeção de dados na escrita a perda de confiabilidade é cerca de 95%.

Tabela 2.2: Síntese do impacto dos nós de má-conduta no PAN

Ataques	Perda nas leituras	Perda nas escritas
Falta de cooperação	5%	10%
Temporização	-	3%
Injeção de dados	73%	95%

Esses ataques tornam o sistema de quórum inviável para gerenciar os serviços de operação de rede em MANETs, necessitando do suporte de mecanismos que garantam a disponibilidade e a integridade dos dados. Existem sistemas de replicação que são tolerantes a esses nós de má-conduta, como o PAXOS [17] e o sistema de quórum bizantino [8]. Entretanto, esses sistemas sustentam fortes premissas, como a garantia de entrega das mensagens e o uso de servidores estáticos, o que é difícil de garantir nas MANETs devido às suas características. Além disso, o PAXOS oferece a tolerância à nós de má-conduta pela realização do consenso das operações, aplicado à replicação de máquina de estados, que não é aconselhável para as MANETs devido à sobrecarga resultante da quantidade de mensagens trocadas [18].

Desta forma, os sistemas de quórum para MANETs precisam ser seguros contra os nós egoístas e maliciosos e preferencialmente conservar as suas características, como a autonomia, a auto-organização e a baixa quantidade de mensagens trocadas. Para isso, eles necessitam de uma solução modular que considere as características dessas redes. Existem várias soluções propostas para a detecção de nós de má-conduta em MANETs, contudo, a maioria deles possui uma alta taxa de troca de mensagens ou consideram autenticação e o uso de entidades centrais. Uma relação desses sistemas é apresentada no próximo capítulo.

2.4 Resumo

Esse capítulo apresentou os sistemas de quórum como um modelo de replicação adequado para gerência de dados de operação de rede em MANETs. Foi explicado o funcionamento dos sistemas de quórum bizantinos e probabilísticos, destacando o uso dos sistemas de quórum probabilísticos em MANETs. Também foram abordados os problemas que os nós egoístas e maliciosos podem causar nas escritas e nas leituras do sistema de quórum PAN. Além disso, verificou-se que esse sistema de quórum é vulnerável aos nós de má-conduta, em especial aos nós maliciosos, e que precisam de uma contra medida para lidar com a presença desses nós nas operações de replicação.

CAPÍTULO 3

MECANISMOS DE TOLERÂNCIA À MÁ-CONDUTA EM MANETS

Este capítulo apresenta os principais trabalhos propostos para a tolerância à má-conduta em MANETs. A Seção 3.1 introduz as técnicas mais comuns empregadas na detecção de nós egoístas e maliciosos em MANETs, relacionando principalmente os sistemas de reputação e os sistemas de identificação de injeção de dados falsos. A Seção 3.2 introduz os conceitos biológicos dos mecanismos de sensoramento em quórum e de seleção por parentesco, que são processos presentes nas bactérias e que inspiram a solução proposta neste trabalho.

3.1 Detecção de nós egoístas e maliciosos

Na literatura, existem diversos mecanismos propostos para a detecção de nós egoístas em MANETs, como os mecanismos de confiança, os de reputação e os baseado em créditos [48]. Em geral, os mecanismos de confiança e de reputação consideram o comportamento dos nós para detectar a má-conduta, isolando aqueles com baixa confiança ou reputação. Já nos mecanismos baseados em créditos, os nós estabelecem uma forma de pagamento para usar os serviços da rede, desencorajando os nós a realizar um comportamento egoísta. Dentre esses mecanismos, o de reputação é o mais empregado nos sistemas em MANETs, por ser distribuído e não depender de entidades centrais [19].

Já para a detecção de nós maliciosos que injetam dados falsos, os nós realizam a autenticação dos dados ou a validação sobre o seu envio. Os sistemas que se baseiam na autenticação devem manter uma infraestrutura de suporte que gerencie essa autenticação, sendo que a maioria dessas infraestruturas considera que a gerência é realizada externamente à rede. Além disso, a detecção de injeção de dados falsos pela validação dos dados exige o estabelecimento de um acordo entre um grupo de nós no envio de um dado para um repositório. Alguns desses sistemas de detecção de nós egoístas e maliciosos são apresentados com maiores detalhes a seguir.

3.1.1 Sistemas de reputação

O esquema CONFIDANT (*Cooperation Of Nodes Fairness In Dynamic Ad-hoc NeTworks*) é aplicado como um componente aos protocolos de roteamento para MANETs, e baseia a reputação dos nós no monitoramento dos seus vizinhos, por meio da técnica de *watchdogs* [20]. Esse esquema detecta a falta de cooperação dos nós, e quando o valor da

reputação de um nó fica abaixo de um determinado limite, o nó é excluído do roteamento. Uma das desvantagens desse esquema é a troca de informações sobre os nós, permitindo que eles enviem acusações falsas sobre os outros nós. Além disso, a técnica de *watchdog* bloqueia os canais de comunicação durante o monitoramento do canal, o que pode gerar um atraso na transmissão dos pacotes. Essa técnica também é vulnerável pois não diferencia um nó malicioso que não encaminha os pacotes de um nó que tenha apresentado uma colisão ocasional no envio dos pacotes.

O sistema SCAN (*Self-organized network-layer security in mobile Ad hoc Networks*) concentra a mitigação de ataques egoístas no nível de roteamento e de encaminhamento de pacotes [21]. Nesse sistema, os nós analisam o comportamento do encaminhamento dos pacotes dos seus vizinhos e detectam a má-conduta deles. Isso é feito através do monitoramento do canal e da comparação dos pacotes enviados e encaminhados pelos vizinhos. Porém, esse sistema impõe uma sobrecarga por conta da troca de informação de reputação e da realização da validação das informações, além do bloqueio dos canais para a verificação do encaminhamento dos pacotes pelos outros nós.

O sistema ACACIA (*A Controller-node-based Access-Control mechanism for Ad hoc networks*) é empregado no controle de acesso dos nós à rede. Para isso, ele utiliza a relação de confiança dos usuários, sendo que somente os usuários convidados podem participar da rede [22]. Por meio de uma cadeia de delegação, são distribuídos convites de modo que os nós convidem outros a fazerem parte da rede. Os donos dos convites são responsáveis pelos seus convidados, e qualquer indício de má-conduta dos nós convidados é reportado ao nó que os convidou. Esse sistema também possui uma sobrecarga de mensagens, visto que os nós controlam a entrega de convites e enviam mensagens ao detectar um comportamento de má-conduta.

Percebe-se que esses sistemas, embora propostos para MANETs e atendendo às suas características, são muito restritos na forma da análise de um comportamento do nó. Uma das características comum é a preocupação com a opinião de vários nós a respeito do comportamento de má-conduta detectado, o que gera um atraso na detecção desses nós. Além disso, a maioria desses sistemas são voltados apenas para a detecção de nós egoístas, não lidando com a detecção de nós maliciosos.

3.1.2 Identificação de injeção de dados falsos

Os autores em [49] propõem a mitigação da injeção de dados falsos em uma rede de sensores por meio do posicionamento estratégico de *firewalls*, e apresentam uma arquitetura para a localização do atacante. Nessa arquitetura, o ataque de injeção de dados é caracterizado por uma grande quantidade de recebimento de pacotes, e os nós detectam os atacantes através da quantidade de dados trafegados. Os nós vítimas de um atacante comunicam o ataque à uma estação base, pertencente a uma rede celular. Essa entidade

recolhe evidências do ataque e exclui o nó atacante da rede. O *firewall* é posicionado em um nó que faça parte da maior quantidade de rotas a serem protegidas, e cujas rotas não possuam nós atacantes. O principal obstáculo para a aplicação desse sistema em MANETs é a necessidade de uma infraestrutura complementar à rede de sensores para o controle da exclusão dos nós.

O protocolo de roteamento EASY [50], também proposto para rede de sensores, tem como principal característica a mitigação de injeção de rotas falsas por meio da assinatura e verificação probabilística dos pacotes, de forma individual e em grupo. Além disso, o EASY baseia-se na técnica de *watchdog* para a identificação dos nós maliciosos e emprega a validação das operações para a sua exclusão. A assinatura e a verificação probabilística dos pacotes diminuem o número de mensagens na rede, contudo, ainda é necessária a troca de mensagens de validação para excluir um nó de má-conduta.

O esquema proposto em [51] visa mitigar a injeção de dados falsos na rede utilizando a colaboração de vários nós no envio de dados. Dessa forma, o esquema tolera um determinado limite de nós maliciosos atuando na rede. Esse esquema cria uma hierarquia de grupos, de forma que a autenticação dos dados é realizada seguindo a hierarquia, identificando de forma mais rápida os nós maliciosos. Entretanto, o esquema exige uma infraestrutura de autenticação dos nós que participam dos grupos, e isto gera uma sobrecarga de mensagens na rede.

Considerando as características dos sistemas de reputação e de identificação de injeção de dados falsos apresentados, percebe-se que em geral eles utilizam entidades externas e centrais e trocam uma grande quantidade de mensagens, características a serem evitadas em MANETs. Desta forma, existe a necessidade de um sistema que identifique os nós de má-conduta de forma *autônoma* e *auto-organizada*, com *pouca troca de mensagens*, e sem dependência de mecanismos de terceiros.

3.2 Mecanismo bio-inspirado para tolerância a ataques

O universo da Biologia provê diversos padrões de comportamento que podem ser instanciados para a área da computação devido à semelhança entre as entidades biológicas e as entidades computacionais. Essa relação possibilita o desenvolvimento de componentes bio-inspirados para a resolução de problemas computacionais, correlacionando o vínculo entre tais entidades. Dessa forma, diversos problemas na área da computação, tais como segurança e desempenho, têm sido modelados e resolvidos com soluções inspiradas em comportamentos biológicos [52]. Esses mecanismos biológicos são classificados em três níveis: filogênese, ontogênese e epigênese. A filogênese se refere à evolução de indivíduos, enquanto que a ontogênese e a epigênese tratam do desenvolvimento por meio do código genético e do aprendizado, respectivamente [53]. Os sistemas computacionais bio-inspirados também podem ser classificados nesses níveis, inclusive podendo participar de

mais de um deles.

Observa-se que as MANETs em particular têm sido amplamente beneficiadas pelo uso de sistemas bio-inspirados na resolução de problemas. O comportamento das formigas, por exemplo, tem servido de inspiração para diversos modelos de protocolos de roteamento [54, 55], enquanto que modelos inspirados no sistema imunológico humano são constantemente empregados na detecção de ataques de naturezas diversas nesse tipo de rede [56]. Outros exemplos de modelos biológicos utilizados como inspiração para a computação são a sincronização de vagalumes [57] e as redes fisiológicas [58]. O modelo de sincronização de vagalumes é empregado em protocolos de sincronização, enquanto que as redes baseadas na fisiologia humana são aplicadas em soluções para a economia de energia para o melhor escalonamento de recursos.

A medida que o campo da Biologia avança no entendimento do comportamento de entidades e sistemas, novos modelos bio-inspirados tendem a ser desenvolvidos na área de computação [52]. Neste contexto, a comunicação das bactérias, também chamada de sensoramento em quórum (*quorum sensing*), é um tópico recente de pesquisa na área da biologia [59], e consequentemente, na computação. O mecanismo de sensoramento em quórum existe para que as bactérias sejam capazes de auto-organizar reações baseadas na quantidade de bactérias presentes no ambiente. Recentemente foi descoberto que esse mecanismo é capaz de sobreviver a bactérias egoístas [60]. Isso é possível através do mecanismo de seleção por parentesco (*kin selection*) [61], por permitir que as bactérias deem preferência à interagir com aquelas que compartilham o mesmo material genético, excluindo as bactérias mutantes.

Os processos de sensoramento em quórum e de seleção por parentesco são uma abordagem promissora para a exclusão de nós egoístas e maliciosos em ambientes de rede em que os nós podem se comportar de forma semelhante às bactérias egoístas, como é o caso das MANETs. Isso porque o processo de sensoramento em quórum ocorre de forma *autônoma*, em que as bactérias verificam o ambiente individualmente, o que é uma característica desejável para as aplicações nas MANETs. Além disso, as bactérias iniciam ações em conjunto de forma *auto-organizada* e o processo de seleção por parentesco favorece a cooperação entre os nós bons, o que promove a tolerância a nós egoístas e maliciosos nas operações da rede. Uma descrição detalhada do sensoramento em quórum e da seleção por parentesco é apresentada a seguir.

3.2.1 Sensoramento em quórum (*Quorum Sensing*)

O sensoramento em quórum é um mecanismo de comunicação entre células, em que as bactérias se baseiam na produção e na detecção de produtos químicos extracelulares para monitorar a densidade de bactérias no ambiente. Esses produtos químicos são chamados de autoindutores, e além de servir como um sinalizador da quantidade de bactérias pre-

sentes no ambiente, também são responsáveis por fornecer nutrientes para o crescimento e a reprodução das bactérias [59]. Ao sensoriar uma determinada quantidade de bactérias no ambiente, elas iniciam um comportamento em conjunto. Deste modo, o sensoriamiento em quórum permite que as bactérias, seres unicelulares e independentes, desenvolvam um comportamento multicelular orientado a evento que torna-se vantajoso para o grupo.

Muitas ações desempenhadas pelas bactérias, como patogenias e criação de biofilmes, não são vantajosas se desempenhadas independentemente. Esse é o caso das doenças causadas por bactérias, como a fibrose cística causada pela bactéria *Pseudomonas Aeruginosa* [62]. Ao agirem sozinhas, as bactérias estão vulneráveis ao sistema imunológico humano, que as destrói antes que atinjam uma quantidade suficiente para causar algum dano ao hospedeiro. O sensoriamiento em quórum permite então que elas aguardem a existência de uma quantidade adequada de bactérias no ambiente para sobrepor o mecanismo de defesa do hospedeiro. Esse princípio também é aplicado às bactérias probióticas, que são as bactérias boas que fazem a proteção dos hospedeiros.

O processo de sensoriamiento envolve a produção e a contagem de autoindutores no ambiente. Cada bactéria possui um receptor, que se liga somente com autoindutores específicos que dependem do tipo de bactéria. Além do receptor, cada bactéria possui genes específicos que se manifestam somente na presença de determinada densidade populacional. A taxa de produção de autoindutores por cada bactéria é baseada na condição da quantidade de bactérias presentes no ambiente. Para isso, elas são divididas em dois tipos: as bactérias que ainda não atingiram a quantidade necessária de autoindutores para modificar o gene controlado pelo sensoriamiento, e as que já atingiram o limite necessário para a mudança do gene. Cada uma possui uma taxa diferente de produção de autoindutores. Já a concentração de autoindutores necessária para desencadear a mudança de genes nas bactérias depende da taxa de reprodução das bactérias, que influencia a produção de autoindutores, e da cooperação entre produção de autoindutores e bactérias. As equações referentes ao cálculo de autoindutores, reprodução de bactérias e concentração de autoindutores são encontradas em [63].

A determinação da quantidade de bactérias presente no ambiente acontece através da ligação dos autoindutores com receptores presentes na membrana celular das bactérias. Quando a quantidade de autoindutores atinge um determinado limite, os genes específicos são ativados, dando início a um comportamento em conjunto. A Figura 3.1 ilustra um ambiente em que as bactérias ativam o gene da bioluminescência através do sensoriamiento em quórum, como acontece com a bactéria *Vibrio fischeri* [64], que só emite luz na presença de um determinado número de bactérias no ambiente. O instante I apresenta um ambiente com poucos autoindutores, e não há a concentração necessária para a ativação do gene da bioluminescência. Conforme as bactérias crescem e se multiplicam, mais autoindutores são emitidos no ambiente, como mostra o instante II. Dessa forma, a quantidade de autoindutores no ambiente aumenta, e através do sensoriamiento em quórum,

as bactérias identificam essa densidade e iniciam a emissão de luz.

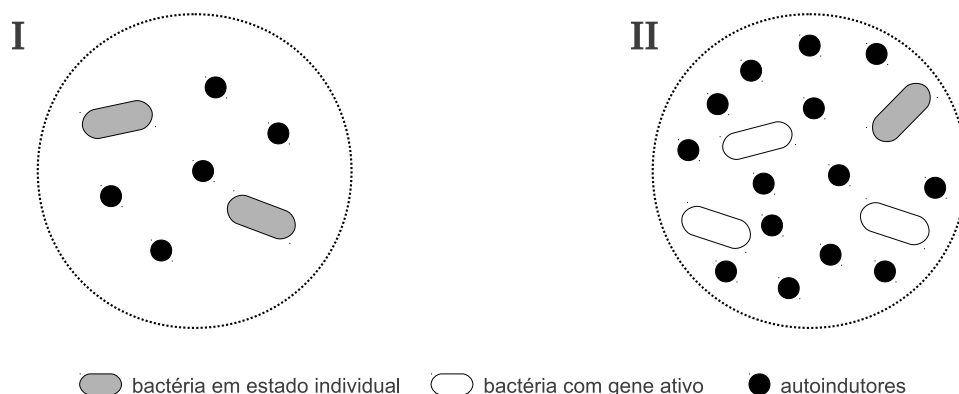


Figura 3.1: Processo de sensoriamento em quórum e início de uma ação em conjunto

Embora o sensoriamento em quórum forneça a melhor estratégia para uma ação em conjunto, ele é custoso do ponto de vista metabólico, visto que as bactérias gastam energia para produzir os autoindutores e os repor no ambiente. Deste modo, o sensoriamento em quórum pode potencialmente ser invadido por bactérias egoístas que se beneficiam dos autoindutores, mas que não têm o custo metabólico de produção, ou ainda bactérias que não participam do comportamento coletivo em resposta à concentração de bactérias no ambiente. Além disso, o sensoriamento em quórum está exposto a bactérias que modificam as propriedades químicas dos autoindutores, com o objetivo de desarticular o sensoriamento da densidade de bactérias. Esse comportamento é chamado de tragédia dos comuns, e mostra que enquanto grupo, a estratégia mais vantajosa é colaborar. Porém esse comportamento não é estável, pois enquanto indivíduo, a melhor estratégia é usufruir do esforço dos demais sem contribuir com o seu próprio esforço [65].

Contudo, sabe-se que as bactérias optam pelo altruísmo, beneficiando o coletivo em detrimento do ganho individual, mesmo na presença de bactérias egoístas [60]. A teoria mais aceita para esse comportamento é a seleção por parentesco. Ela afirma que ao ajudar indivíduos cujo material genético é compartilhado, as bactérias passam indiretamente os seus genes para a próxima geração, garantindo a continuidade da espécie [61]. Assim, a cooperação acontece preferencialmente entre bactérias que possuem parentesco. Essa seleção acontece por discriminação das bactérias que não possuem o mesmo comportamento ou pela dispersão limitada das bactérias que compartilham os mesmos genes. Nesse trabalho, foca-se na exclusão de entidades que não apresentam o mesmo comportamento, que é obtida pela seleção por parentesco, abordada a seguir.

3.2.2 Seleção por parentesco (*Kin Selection*)

A seleção por parentesco é um processo fundamental da evolução. Essa teoria tem como base a regra de Hamilton, que afirma que a cooperação altruísta entre dois indivíduos

é favorecida quando eles são relacionados geneticamente, e depende do benefício para o indivíduo a ser ajudado e do custo de ajuda do indivíduo altruísta [65]. A relação de parentesco entre dois indivíduos torna-se uma variável primordial para a decisão de cooperação entre eles. Ela expressa a probabilidade do gene que causa o altruísmo estar presente em ambos os indivíduos, e isso determina a semelhança genética e a probabilidade de ajuda mútua entre eles.

Com essa relação de parentesco, as bactérias favorecem a ajuda às bactérias que fazem parte da mesma linhagem genética, criando um mecanismo de combate à bactérias que não cooperam com a difusão dos autoindutores e com a ação em conjunto, disparada pelo sensoramento em quórum. Uma outra explicação para a sobrevivência do sensoramento em quórum na presença de bactérias egoístas é o mutualismo, em que as bactérias ajudam aquelas que as ajudam, contudo a teoria da seleção por parentesco é a mais aceita [65].

O sensoramento em quórum e a seleção por parentesco são mecanismos dinâmicos e independentes, pois as bactérias utilizam os autoindutores para seu crescimento e aproveitam esse produto para a realização de uma ação em conjunto. Esse comportamento é interessante para o ambiente das MANETs, em que os nós apresentam um comportamento dinâmico e independente dos demais nós da rede. Além disso, os nós podem usar as próprias mensagens enviadas como autoindutores, o que não gera custos para a produção e envio de autoindutores.

3.3 Resumo

Esse capítulo apresentou alguns dos mecanismos de reputação e de identificação de injeção de dados falsos propostos para a identificação de nós de má-conduta, descrevendo suas funcionalidades e vulnerabilidades. Além disso, destacou-se a necessidade de uma solução que englobe a detecção de nós egoístas e maliciosos em conjunto, e relacionou as características desejáveis em um sistema de detecção para MANETs. Esse capítulo também introduziu os mecanismos de sensoramento em quórum e de seleção por parentesco, que são mecanismos utilizados pelas bactérias para a organização de ações em conjunto e para a exclusão de bactérias egoístas do ambiente em que vivem. Esses dois mecanismos formam a base da solução proposta para a tolerância de nós egoístas e maliciosos nos sistemas de quórum para MANETs, abordada no Capítulo 3.

CAPÍTULO 4

UM ESQUEMA BIO-INSPIRADO PARA A TOLERÂNCIA DE NÓS DE MÁ-CONDUTA

Este capítulo apresenta um esquema bio-inspirado para tolerância a nós de má-conduta em sistemas de quórum para MANETs. O esquema proposto, denominado QS^2 , é inspirado nos mecanismos de sensoriamento em quórum e de seleção por parentesco encontrados nas bactérias, e são aplicados na exclusão de nós de má-conduta das operações de um sistema de quórum. A Seção 4.1 apresenta uma visão geral da arquitetura e das características do esquema QS^2 , assim como as asserções consideradas. A Seção 4.2 associa as entidades biológicas e as entidades computacionais e explica suas funções. A Seção 4.3 apresenta o funcionamento do QS^2 e exemplifica o seu uso em operações de escrita e de leitura de dados em sistemas de quórum para MANETs.

4.1 Visão geral do esquema QS^2

O esquema QS^2 (*quorum system + quorum sensing*) tem como objetivo proporcionar um meio para a criação de quórums de escrita e de leitura com participantes colaborativos, e negar a participação de nós de má-conduta nas operações dos quórums. Para isso, os nós realizam a classificação e a exclusão de nós com base nos mecanismos de sensoriamento em quórum e de seleção por parentesco, ambos empregados pelas bactérias. No QS^2 , cada nó tem uma visão individual do comportamento dos outros nós na rede, que depende da qualidade da interação entre eles. Além disso, cada nó toma decisões de acordo com sua própria experiência sobre os demais nós. O esquema é composto por dois módulos: o módulo de monitoramento dos nós e o módulo de decisão de cooperação, conforme ilustrado na Figura 4.1.

O *módulo de monitoramento dos nós* é responsável pela classificação dos nós em confiáveis ou de má-conduta. Esse módulo é subdividido em dois componentes: o monitoramento do comportamento e a classificação dos nós. O monitoramento do comportamento dos nós computa a quantidade de autoindutores enviados por cada nó da rede. Os autoindutores para o QS^2 são as escritas enviadas e os encaminhamentos de dados realizados por cada nó na rede. A classificação dos nós relaciona cada nó com um dos três estados: confiáveis, egoístas ou maliciosos. Isso depende da contagem de autoindutores de cada nó e dos limites dos autoindutores estabelecidos.

O *módulo de decisão de cooperação* determina a relação de cooperação entre dois nós, selecionando para interação somente aqueles que tem maior probabilidade de ajudar. Esse

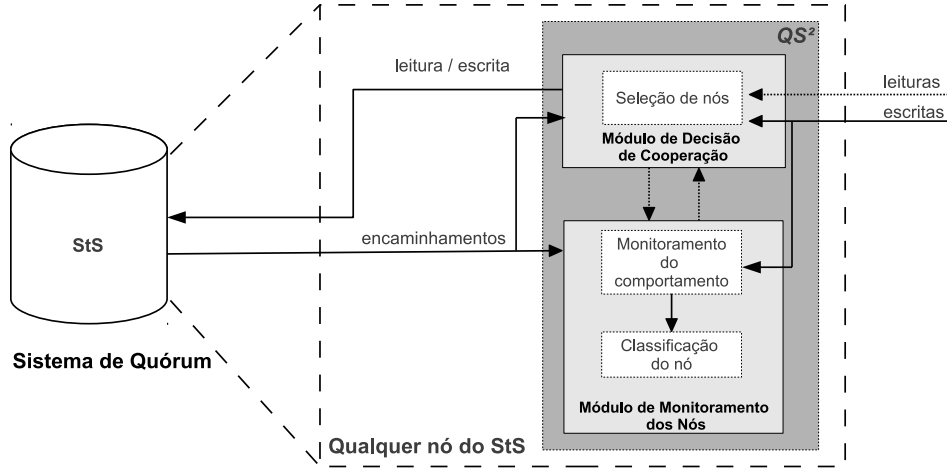


Figura 4.1: Arquitetura do esquema QS^2

módulo é composto pelo componente de seleção de nós, que é apoiado pelo módulo de monitoramento dos nós. Esse módulo também é capaz de tornar a seleção mais flexível e adequar a seleção e a interação entre os nós. Deste modo, pode-se considerar um nó bom para uma determinada operação e egoísta ou malicioso para outras. Em conjunto, os módulos de monitoramento dos nós e de decisão de cooperação determinam quais nós são confiáveis, isto é, nós cujo comportamento é colaborativo. Tais nós são posteriormente escolhidos para participar dos quóruns de escrita e de leitura.

O QS^2 é *autônomo*, pois contabiliza individualmente a relação de colaboração entre os nós, e é *auto-organizado* devido à forma com que os nós iniciam a exclusão de outros nós de má-conduta sempre que esses atingem determinada quantidade de escritas ou encaminhamentos. Além disso, cada nó tem uma visão independente do comportamento dos nós na rede. O QS^2 não necessita de mensagens extras para o envio de autoindutores, o que o torna uma solução com *baixo custo de comunicação*. Ele também não requer pontos fixos de controle e nem depende de outros esquemas de reputação. Apesar da independência dos nós no uso do QS^2 , os nós confiáveis decidem pela mesma ação ao detectar um nó de má-conduta, sendo que os nós de má-conduta agem arbitrariamente com relação ao QS^2 .

4.1.1 Modelo do sistema

O sistema considera uma rede MANET composta por dispositivos móveis, como *smartphones* e *notebooks*, capazes de se comunicar por rádio-frequência. A comunicação entre os nós acontece quando eles estão localizados no raio de transmissão um do outro, ou por meio da ajuda de outros nós, através do roteamento multissalto. Considera-se que o QS^2 seja aplicado em sistemas de quórum probabilísticos para MANETs. O sistema de quórum apoia a replicação de dados de serviços de operação de rede, tais como as informações de localização e de conectividade. Assume-se também que os nós da MANET se movimen-

tam seguindo um padrão de movimentação em uma área delimitada. Esse cenário pode caracterizar uma rede composta por pedestres, ciclistas ou carros em uma área urbana com algum tipo de dispositivo sem fio, ou ainda um grupo de pessoas equipadas com dispositivos móveis em um evento. O modelo do sistema é composto de três camadas: o **modelo de rede**, o **modelo de gerência de dados** e o **modelo de ataque**. O modelo em camadas é ilustrado na Figura 4.2.

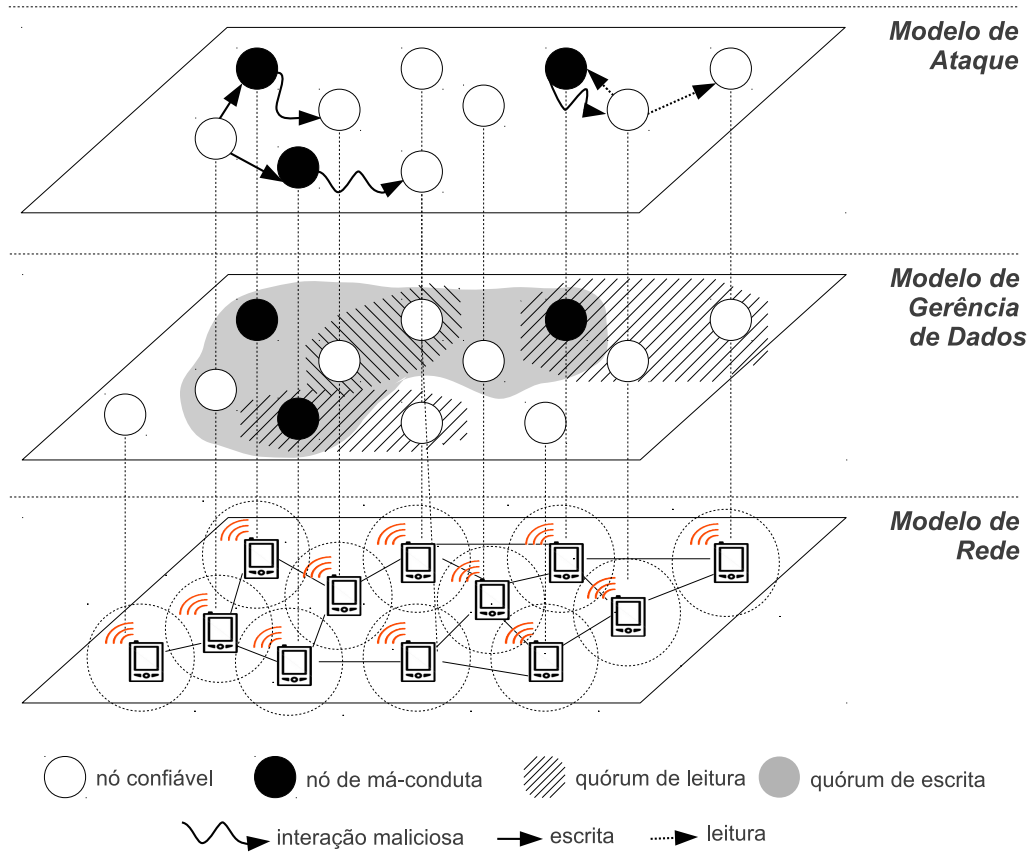


Figura 4.2: Modelo do sistema em camadas

Modelo físico da rede - Para a execução do QS^2 , assume-se que a rede é formada por um conjunto P composto por n nós identificados por $\{s_0, s_1 \dots s_{n-1}, s_n\}$, sendo que todo nó $s_i \in P$ tem um endereço físico ou identificador único. Esses nós se comunicam através de um canal sem fio, cujo raio de transmissão é igual para todos os nós da rede. Considera-se que os processos e os canais de comunicação são assíncronos, isto é, o tempo de transmissão é variável e desconhecido. O canal de comunicação não é confiável, e está sujeito a perda de pacotes devido a colisão ou a entrada e saída de nós. Além disso, os nós podem se movimentar e modificar o raio de alcance da antena de transmissão, o que também pode gerar perda de pacotes.

A rede pode ser particionada pela movimentação ou pela entrada e saída dos nós. Além disso, os nós não possuem conexão com todos os outros, e deste modo, as mensagens precisam ser roteadas por meio de nós intermediários até o destino. Supõe-se que o

roteamento é seguro, sendo que nós de má-conduta não podem comprometer a descoberta e a manutenção das rotas. Da mesma forma, assume-se que as mensagens de replicação são relativamente pequenas com no máximo 128 *bytes*. Essa característica é comum nos serviços de operação da rede que, em geral, enviam mensagens pequenas para o controle e o monitoramento da rede. Portanto, elas podem ser enviadas em pacotes únicos. Além disso, assume-se que a rede fornece um esquema de assinatura, tais como esquemas de assinaturas baseado em identidade [66, 67, 68], para a proteção de informações importantes enviadas pelo QS^2 , de forma que nós de má-conduta não possam modificar tais informações.

Modelo de gerência de dados - O gerenciamento de dados é realizado por um sistema de quórum probabilístico. Neste trabalho, foi escolhido o PAN [9], embora outros sistemas de quórum probabilísticos possam ser aplicados. O PAN é composto por um sistema de armazenamento (StS) e por nós clientes, servidores e agentes. O StS é composto por nós servidores. Os nós servidores armazenam os dados e gerenciam a replicação entre eles. Os nós clientes requisitam dados para os servidores. Os servidores que são contatados diretamente pelos clientes são denominados agentes, e são responsáveis por mediar as requisições de leitura e de escrita entre os clientes e o StS. A escrita é baseada em um protocolo epidêmico, sendo que a sua disseminação entre os nós acontece em intervalos regulares. Já a leitura é realizada no quórum de leitura de tamanho pré-determinado, e é feita por meio de mensagens *unicast*. Esse sistema de quórum foi escolhido por conta do uso de quóruns assimétricos, reduzindo o número de mensagens na replicação, e ser voltado para a replicação de dados pequenos, como dados de serviços de operação de rede. Além disso, o PAN considera as características específicas das MANETs, e emprega mecanismos que facilitam a gerência do armazenamento dos dados em redes dinâmicas como as MANETs.

Modelo de falhas - O QS^2 trata de nós de má-conduta que afetam a propriedade da disponibilidade e da integridade dos dados em um sistema de replicação. Esses nós de má-conduta são intrusos e conhecem o funcionamento da rede, tendo permissão e chaves criptográficas para participar das operações. Assume-se que um nó s_i é egoísta se ele não colabora com as operações de replicação dos sistemas de quórum, e malicioso se ele modifica ou injeta dados maliciosos no sistema de replicação. Um nó s_i pode ser egoísta ou malicioso, ou apresentar ambos os comportamentos ao mesmo tempo. Considera-se dois tipos de comportamento para os nós maliciosos: a injeção de dados falsos e o atraso na propagação dos dados. Já os nós egoístas não cooperam com as operações do sistema de quórum. Assume-se que um nó de má-conduta se comporta de modo egoísta ou malicioso durante toda a sua participação na rede, de modo que o comportamento de má-conduta não seja intermitente.

4.2 Entidades bio-inspiradas

O esquema QS^2 contabiliza as escritas e os encaminhamentos para indicar a qualidade da interação dos nós na rede, assim como é observado no mecanismo de sensoriamento em quórum. O esquema assume dois tipos de autoindutores: os autoindutores associados às escritas, denominados de $AI-W$, e os autoindutores relacionados aos encaminhamentos de escritas, denominados de $AI-F$. Ao enviar escritas e encaminhar dados, os nós acrescentam a rota de disseminação pela qual o dado foi disseminado, de modo que os próximos nós contabilizem a sua colaboração na rede. Todos os nós possuem contadores individuais para os autoindutores $AI-W$ e $AI-F$. Os nós que recebem uma escrita incrementam a quantidade de $AI-W$ para o nó de origem e a quantidade de $AI-F$ para cada nó contido na rota de disseminação.

A Figura 4.3 ilustra a disseminação dos autoindutores $AI-W$ e $AI-F$ no QS^2 , em uma rede composta por 7 nós ($\{s_0 - s_6\}$), em que os nós s_1 e s_5 atuam de modo egoísta. Os nós enviam a rota de disseminação através de um campo denominado *rota*. O nó s_0 inicia a disseminação de um dado para os nós s_1 e s_6 , e a medida que o dado é disseminado, os nós acrescentam seus identificadores no campo *rota*, como é o caso dos nós s_6 e s_3 . Os nós s_1 e s_5 , que são egoístas, não repassam o dado para outros nós, e portanto, não possuem o identificador no campo *rota*. Ao receberem o dado enviado por s_0 , os nós atualizam a contagem de $AI-W$ para o nó s_0 , que foi a origem do dado, e de $AI-F$ para cada nó presente na rota de disseminação. Conforme as operações de escrita acontecem na rede, os nós confiáveis incrementam a sua contagem de autoindutores nos outros nós, ao contrário dos nós egoístas e maliciosos em que a contagem de autoindutores acusa um comportamento anormal para a rede.

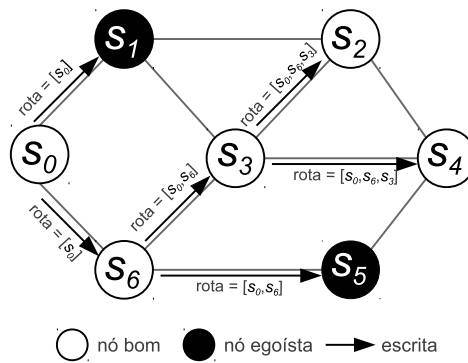


Figura 4.3: Disseminação e contagem de autoindutores no esquema QS^2

Na etapa de monitoramento do comportamento dos nós, o QS^2 compara o comportamento esperado para um nó e o comportamento realmente identificado por meio da contagem dos autoindutores $AI-W$ e $AI-F$. O comportamento de um nó é estabelecido pela combinação de dois genes: o gene **c** e o gene **m**, que representam respectivamente os genes para nós egoístas e nós maliciosos. A Tabela 4.1 apresenta os fenótipos, que são

os comportamentos possíveis para os nós na rede, combinando os genótipos (genes) **c** e **m**. Dessa forma, um nó identificado como egoísta possui o par de genes **Cm**, e um par de genes **cM** caracteriza um nó malicioso. Além disso, a permuta desses genes dá origem a outros fenótipos, como um nó malicioso que também apresenta um comportamento egoísta (**CM**).

Tabela 4.1: Comportamento dos nós na rede

Genótipos	Fenótipos
cm	bom
cM	malicioso
Cm	egoísta
CM	malicioso egoísta

Na classificação dos nós, o esquema QS^2 verifica a quantidade de autoindutores enviada pelos nós e a compara com uma quantia identificada como aceitável para a rede. Para isso, estima-se a taxa esperada de escritas enviadas por um nó, denominada k_{env} , e a taxa de encaminhamentos de escritas, denominada k_{enc} . Ambas as taxas são calculadas em função de um determinado período de tempo. A partir dessas taxas, se determina os limites de envio para os autoindutores $AI-W$ e $AI-F$. Qualquer nó que esteja além desses limites é identificado como um nó de má-conduta. Tais limites são obtidos pela análise da quantidade de escrita e de encaminhamentos que cada nó envia à rede, e depende do tipo do dado replicado e do comportamento dos serviços que utilizam esse dado.

Nesse trabalho, assume-se que o intervalo de envio de escritas e de leituras é definido por uma distribuição de Poisson, devido à adequação dessa distribuição ao comportamento dos dados dos serviços de operação de rede. Contudo, o esquema QS^2 pode considerar outras funções de distribuição de escritas e de leituras. Seguindo essa distribuição, o QS^2 determina que a média de envio de escritas e leituras é igual a λ . A quantidade de escritas enviadas por cada nó é obtida pela soma das probabilidades de envio de x escritas, descrita pela Equação (4.1), onde e equivale a 2,71828. Já a taxa de encaminhamentos dos nós é obtida pela soma das taxas k_{env} de todos os nós, como mostra a Equação (4.2), em que n é a quantidade de nós na rede.

$$k_{env} = \sum_{x=0}^{\infty} x \frac{\lambda^x \times e^{-\lambda}}{x!} dx \quad (4.1)$$

$$k_{enc} = \sum_{i=0}^n k_{env_i} \quad (4.2)$$

A partir dessas taxas, calcula-se os limites de envio de escrita, k_{env}^{max} , e de encaminhamentos, k_{enc}^{min} , considerados normais para os nós. Um nó é malicioso se ultrapassar o limite máximo permitido de escritas durante um determinado período de tempo, e é egoísta se não atingir e sustentar um limite mínimo de escritas encaminhadas.

A taxa máxima de envio de escritas k_{env}^{max} para um nó bom é calculada pela Equação (4.3), em que δ representa a probabilidade do envio de escritas ser menor do que o

$k_{env^{max}}$ estimado. Por exemplo, configurando o $\delta = 0,90$, obtêm-se uma probabilidade de 10% para que um nó envie mais de $k_{env^{max}}$ escritas.

$$\sum_{i=0}^{k_{env}} \frac{\lambda^{k_{env^{max}}} \times e^{-\lambda}}{k_{env^{max}}!} \leq \delta \quad (4.3)$$

A quantidade mínima de encaminhamentos para um nó é calculada pela Equação (4.4), em que γ representa a probabilidade dos nós encaminharem menos de $k_{enc^{min}}$. Por exemplo, $\gamma = 0,5$ significa uma probabilidade de 5% que um nó encaminhe menos do que $k_{enc^{min}}$ dados. Os nós egoístas e maliciosos possuem taxas k_{env} e k_{enc} arbitrárias, e não respeitam as taxas $k_{env^{max}}$ e $k_{enc^{min}}$ definidas pelo esquema.

$$\sum_{i=0}^{k_{enc}} \frac{\lambda^{k_{enc^{min}}} \times e^{-\lambda}}{k_{enc^{min}}!} \geq \gamma \quad (4.4)$$

Depois do monitoramento do comportamento e da classificação do nó, o módulo de decisão de cooperação seleciona os nós que podem participar das operações do sistema de quórum. Essa decisão está baseada nos genes identificados pela etapa de determinação dos genes do nó e pelo tipo de operação que o nó deseja realizar. A operação de leitura pode admitir a escolha de um nó egoísta para compor o quórum de leitura. Isso porque a leitura conta com mais nós em um quórum e a má-conduta egoísta de um componente não prejudica de forma acentuada o andamento da operação. Porém, isso não é possível em uma operação de escrita, em que um nó egoísta compromete por completo a propagação de um dado. Sendo assim, um nó pode decidir cooperar com outros baseando-se nos genes determinados pelas etapas anteriores e na operação que ele deseja realizar.

4.3 Funcionamento do QS^2

A identificação e a contabilização dos autoindutores no esquema QS^2 acontece por meio de informações enviadas juntamente com as mensagens das operações de replicação. Essas informações definem os identificadores dos nós de origem e destino da operação, o tipo de operação, a rota da disseminação do dado e o seu valor. A Figura 4.4 ilustra os campos de informações que são adicionados às mensagens de escrita.

ori	dst	tipo op	rota
dado			timestamp


 campo criptografado

Figura 4.4: Informações enviadas pelo QS^2 nas operações de leitura e de escrita de dados

O campo **origem** é preenchido pela origem do dado, que assina este campo de modo que um nó não envie dados sem ser identificado. Uma vez preenchido, esse campo é imutável. O campo **destino** é preenchido pelo nó que realiza a operação, e muda conforme o pacote é enviado pela rede, no caso das operações de escrita. O campo **tipo de operação** identifica o tipo de operação realizada, que pode ser uma escrita ou uma leitura. O campo **rota** armazena a rota de disseminação do dado e também é assinado pelos nós, de modo que nós de má-conduta não modifiquem o conteúdo. A operação de leitura não preenche esse campo, visto que a leitura é realizada em *unicast*. Finalmente, o campo **dado** possui o identificador e o valor do dado, e é assinado para impedir a modificação do seu conteúdo.

4.3.1 Operações

As operações do QS^2 estão divididas em três algoritmos que correspondem às etapas do QS^2 dentro dos módulos de monitoramento dos nós e de decisão de cooperação. O Algoritmo 1 detalha o monitoramento do comportamento dos nós através da contagem dos autoindutores $AI-W$ e $AI-F$ pelo nó s_i . Cada nó executa esse algoritmo no momento em que recebe uma requisição de escrita de um dado. O nó s_i identifica e inicializa os nós que estão interagindo com ele pela primeira vez (l.3). Depois de inicializados, o nó incrementa seu contador de $AI-F$ para cada nó na rota de disseminação (l.4) e a quantidade de $AI-W$ relacionado ao nó de origem (l.6).

Algoritmo 1 Monitoramento dos nós pelo nó s_i

```

1: procedimento MONITORANO()
2:   para todos  $noRota \in rota$  faça
3:      $tempo[noRota] \leftarrow tempoAgora$ 
4:      $qtdeAIF[noRota] \leftarrow qtdeAIF + 1$            ▷ atualiza quantidade do autoindutor  $AI-F$ 
5:   fim para
6:    $qtdeAIW[noOrigem] \leftarrow qtdeAW + 1$          ▷ atualiza quantidade do autoindutor  $AI-W$ 
7: fim procedimento
```

A Figura 4.5 ilustra o monitoramento dos nós no QS^2 . Nela, o nó s_7 inicia a escrita de um dado na rede, enviando a escrita e o seu identificador para dois servidores. Ao encaminhar a escrita, os nós incluem o seu identificador na rota, para que essa colaboração seja contabilizada. A tabela exemplifica a contagem de autoindutores $AI-W$ e $AI-F$ pelo nó s_0 , para os nós da rota $s_7 - s_4 - s_3 - s_2$. O nó s_0 incrementa a quantidade de $AI-W$ para o nó s_7 , a origem do dado, e incrementa a quantidade de $AI-F$ para os nós s_4 , s_3 e s_2 , que encaminharam esse dado até ele.

O Algoritmo 2 descreve a classificação do nó s_w pelo nó s_i . No recebimento de uma escrita enviada pelo nó s_w , o nó s_i primeiramente verifica o tempo que ele está interagindo com a origem, que é o nó s_w (l.2). Caso eles ainda não interagiram, o nó s_i não possui informação suficiente para determinar se o nó s_w é malicioso ou egoísta, e nesse caso, ele considera que o nó s_w é bom (l.3-4). Caso contrário, ele obtém a contagem de $AI-W$

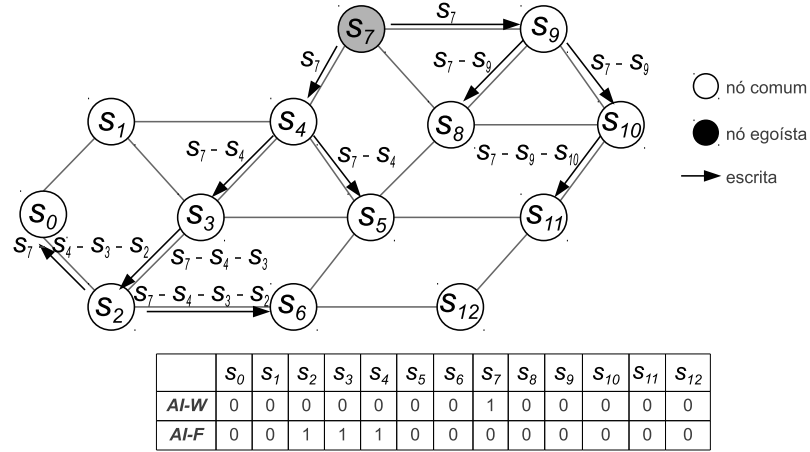


Figura 4.5: Monitoramento dos nós pelo QS^2

e $AI-F$ baseado no tempo em que estão interagindo (l.6-7) e verifica se a quantidade de autoindutores está de acordo com os limites $k_{env^{max}}$ e $k_{enc^{min}}$ definidos pelo esquema. Dessa forma, os nós são classificados como egoístas ou maliciosos, dependendo dos limites definidos e da contagem de autoindutores (l.8-13).

Algoritmo 2 Classificação do nó s_w pelo nó s_i

```

1: procedimento CLASSIFICANO(nó)
2:   se tempo[ $s_w$ ] =  $\emptyset$  então                                     ▷ não é possível classificar o nó
3:      $c \leftarrow 0$ 
4:      $m \leftarrow 0$ 
5:   senão
6:      $cont_{AIW} \leftarrow qtdeAIW[s_w]/tempo[s_w]$                      ▷ calcula a quantidade de escritas enviadas
7:      $cont_{AIF} \leftarrow qtdeAIF[s_w]/tempo[s_w]$                      ▷ calcula a quantidade de escritas encaminhadas
8:     se  $cont_{AIW} > k_{env^{max}}$  então                                   ▷ nó é identificado malicioso
9:        $m \leftarrow 1$ 
10:    fim se
11:    se  $cont_{AIF} < k_{enc^{min}}$  então                                   ▷ nó é identificado egoísta
12:       $c \leftarrow 1$ 
13:    fim se
14:  fim se
15: fim procedimento

```

A Figura 4.6 ilustra a classificação dos nós de acordo com a contagem de autoindutores pelo nó s_0 , conforme demonstra a tabela do nó. A medida que ocorrem as operações de escrita, os nós contabilizam os autoindutores. O nó s_0 classifica os nós s_1 , s_6 , s_8 , s_{11} e s_{12} como egoístas (gene **C**) por não contabilizarem nenhum encaminhamento para ele até o momento. Além disso, o nó s_1 também foi classificado como um nó malicioso (gene **M**), conforme a mostra a tabela, porque enviou mais escritas do que o permitido em um determinado período de tempo. Com esse cenário, o nó s_0 seleciona os nós s_2 e s_3 para enviar um dado, e evita escolher os nós de má-conduta para participar dos quórums.

Depois de classificar o nó s_w , o nó s_i decide se aceita o comportamento do nó s_w . Essa decisão baseia-se no tipo de operação que o nó s_i vai realizar, e é detalhada no

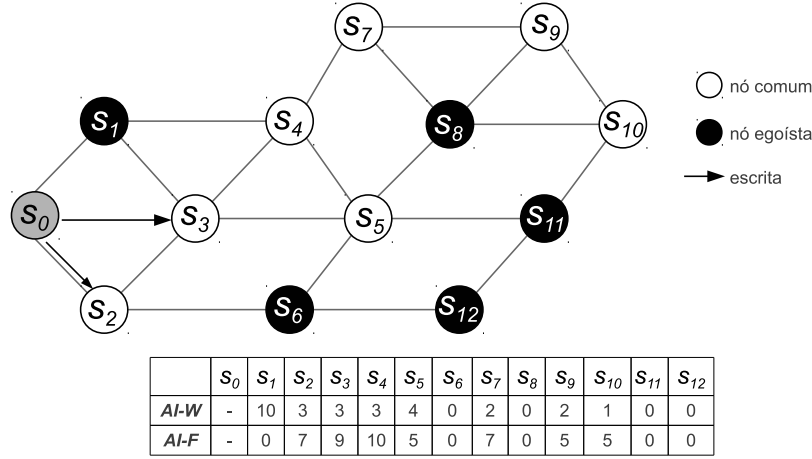


Figura 4.6: Classificação dos nós pelo QS^2

Algoritmo 3. Para as operações de leitura, o QS^2 assume como bom um nó que não é egoísta ou malicioso (l.2-5). Já para as operações de escritas, o nó é considerado bom se não for egoísta nem malicioso (l.6-9).

Algoritmo 3 Decisão de cooperação pelo nó s_i

```

1: procedimento DECIDECOOPERACAO(no)
2:   se  $op = leitura$  então
3:     se  $(m_{s_w} = 0) \vee (c_{s_w} = 0)$  então                                ▷ nó é bom se não for malicioso ou egoísta
4:        $nosBons \leftarrow no$                                               ▷ adiciona nó bom na lista nosBons
5:     fim se
6:   senão                                                                    ▷ operação de escrita
7:     se  $(m_{s_w} = 0) \wedge (c_{s_w} = 0)$  então                                ▷ nó é bom se não for malicioso e nem egoísta
8:        $nosBons \leftarrow s_w$                                               ▷ adiciona nó bom na lista nosBons
9:     fim se
10:  fim se
11: fim procedimento

```

A Figura 4.7 ilustra a execução da decisão de cooperação em operações de escrita e de leitura. O nó s_3 escolhe os nós s_4 , s_5 e s_6 para realizar uma operação de leitura, enquanto que o nó s_9 escolhe os nós s_7 e s_{10} para realizar uma operação de escrita. O nó s_3 escolhe o nó s_6 , apesar de ser identificado como egoísta pela contagem de autoindutores indicada na tabela, porque o algoritmo de decisão de cooperação flexibiliza a seleção para as operações de leitura. Isso é possível porque o nó s_3 pode completar a requisição de leitura corretamente mesmo que o nó s_6 não responda ou modifique essa requisição. Já para as escritas, o QS^2 escolhe somente os nós que sejam bons em ambos os genes, pois a escrita não suporta a interação de nenhum tipo de nó de má-conduta.

A função *selecionaNos(qtdeNos)*, descrita no Algoritmo 4, cria uma lista de nós confiáveis. A partir dessa lista, os nós podem escolher um outro nó para interagir nas operações de escrita e de leitura de um sistema de quórum para MANETs. Primeiramente o nó identifica os genes de um outro nó, conforme descrito na função *determinaGenes(no)*, do Algoritmo 2 (l.3). Em seguida decide se coopera com esse nó por meio da função

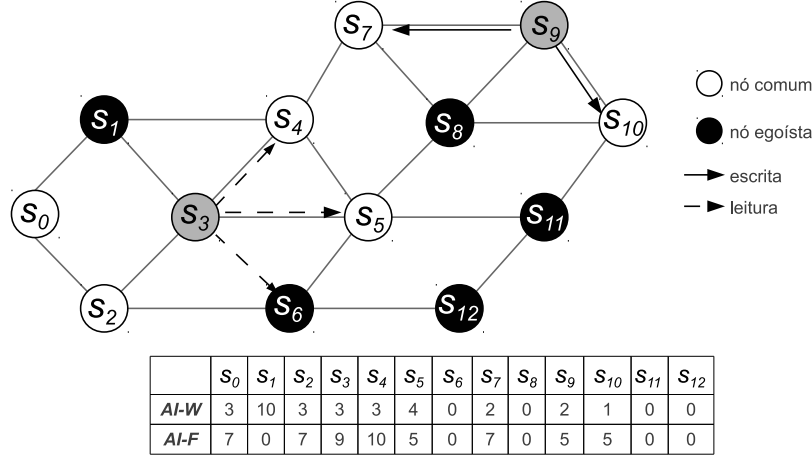


Figura 4.7: Decisão de cooperação pelo QS^2

$decideCooperacao(no)$, do Algoritmo 3 (l.4). Por fim, se o nó atender aos critérios de seleção, ele é inserido na lista de nós confiáveis (l.5).

Algoritmo 4 Seleciona nós

```

1: procedimento SELECIONANós(qtdeNos)
2:   para  $no \in qtdeNos$  faça
3:      $classificaNo(no)$ 
4:      $decideCooperacao(no)$ 
5:      $nosBons \leftarrow no$  ▷ acrescenta o nó na lista de nós confiáveis
6:   fim para
7: fim procedimento

```

4.3.2 Escrita

O procedimento da operação de escrita é apresentado no Algoritmo 5, e inicia com a criação do pacote com todas as informações do cabeçalho (l.2-6). Nas escritas, o QS^2 é empregado pelo nó de origem e pelo nó que recebe uma escrita. O nó de origem da escrita aplica o QS^2 para escolher os nós que vão receber a escrita, e os nós que recebem uma escrita utilizam o QS^2 para aceitá-la ou não, de acordo com o comportamento do nó de origem (l.10-15).

4.3.3 Leitura

O Algoritmo 6 apresenta o procedimento de leitura no QS^2 . A operação de leitura emprega o QS^2 para escolher os nós do quórum de leitura e para determinar o comportamento do nó de origem de um dado recebido como resposta de uma requisição de leitura. Após criar o pacote com as informações necessárias (l.2-6), o nó aguarda pela resposta dos nós do Q_r por um tempo (*timeout*). Os nós do Q_r respondem à requisição se o dado for mais atual do que o recebido (l.10-12). Após o término do *timeout*, o nó verifica se a origem do

Algoritmo 5 Operação de escrita com o QS^2 realizada pelo nó s_i

```

1: procedimento ENVIAESCRITA()                                ▷ nó cliente  $s_i$  enviando uma escrita para o nó  $s_w$ 
2:    $origem \leftarrow s_i$                                        ▷ adiciona o  $id$  do nó de origem
3:    $destino \leftarrow selecionaNos(f)$                          ▷ seleciona nó para enviar o dado
4:    $tipoOp \leftarrow escrita$ 
5:    $rota \leftarrow s_i$                                        ▷ adiciona o  $id$  na rota
6:    $dado \leftarrow dado_{s_i}$                                    ▷ adiciona o novo dado
7:    $enviaPacote$ 
8: fim procedimento
9: procedimento RECEBEESCRITA( $pacote$ )                        ▷ nó  $s_w$  ao receber uma escrita do cliente  $s_i$ 
10:  se  $dadoRecebido.timestamp > dado.timestamp$  então          ▷ verifica se o dado recebido é mais atual
11:     $classificaNo(s_i)$ 
12:    se  $m_{s_i} = 0 \wedge c_{s_i} = 0$  então                        ▷ verifica se o nó origem é malicioso ou egoísta
13:       $dado \leftarrow dadoRecebido$                              ▷ atualiza o dado armazenado
14:    fim se
15:  fim se
16: fim procedimento

```

dado não é maliciosa ou egoísta (l.16-18), e então incrementa os autoindutores da origem do dado e dos nós que estão na rota da disseminação (l.19). Isso é possível porque as respostas encaminhadas pelos nós contém a rota de disseminação desse dado, e portanto, permite a verificação do comportamento do nó de origem.

Algoritmo 6 Operação de leitura com o QS^2 realizada pelo nó s_i

```

1: procedimento ENVIALEITURA                                ▷ nó  $s_i$  emitindo uma requisição de leitura para o nó  $s_w$ 
2:    $origem \leftarrow s_i$                                        ▷ adiciona o  $id$  do nó de origem
3:    $destino \leftarrow selecionaNos(Q_r)$                        ▷ seleciona nó para enviar o dado
4:    $tipoOp \leftarrow leitura$ 
5:    $rota \leftarrow \emptyset$                                      ▷ rota em uma operação de leitura é inexistente
6:    $dado \leftarrow dado_{s_i}$                                    ▷ adiciona o valor do dado armazenado
7:    $enviaPacote$ 
8: fim procedimento
9: procedimento RECEBELEITURA( $pacote$ )                        ▷ nó  $s_w$  ao receber uma requisição de leitura do nó  $s_i$ 
10:  se  $dadoRecebido.timestamp < dado.timestamp$  então          ▷ responde para o nó com o dado atual
11:     $dadoResposta \leftarrow dado$ 
12:     $enviaResposta()$ 
13:  fim se
14: fim procedimento
15: procedimento RECEBERESPONSALEITURA( $pacote$ )                ▷ nó  $s_i$  ao receber a resposta do nó  $s_w$ 
16:    $classificaNo(s_w)$                                        ▷ verifica se a origem do dado é boa
17:   se  $m_{s_w} = 0 \wedge c_{s_w} = 0$  então
18:      $dado \leftarrow resposta$ 
19:      $monitoreaComportamento()$ 
20:   fim se
21: fim procedimento

```

Esse procedimento evita que um nó malicioso fabrique um dado com um *timestamp* atual e envie esse dado como resposta para um nó. Empregando a verificação dos nós na leitura, o QS^2 evita que o nó escolha como resposta um dado enviado por um nó malicioso.

Como descrito anteriormente, o esquema QS^2 enfatiza as características de autonomia,

auto-organização e utilização de poucos recursos, porém tais características implicam em alguns obstáculos para a solução. Por exemplo, a autonomia na detecção de nós de má-conduta pode fazer com que alguns nós que estejam longe sejam detectados como nós egoístas, porque estão com dificuldades temporárias de conexão com os demais nós. Outra barreira é a necessidade de limites estáticos que caracterizem o comportamento de nós confiáveis e de má-conduta. Isso faz com que a escalabilidade do esquema seja insuficiente para determinados sistemas. Além disso, o uso de um esquema de assinaturas impõe um custo de processamento e comunicação na solução.

4.4 Resumo

Esse capítulo apresentou o esquema QS^2 para a exclusão de nós de má-conduta em sistemas de quórum para MANETs. Foram associadas as entidades biológicas no esquema QS^2 e descrito seu funcionamento. Os módulos do QS^2 contabilizam os autoindutores, determinam os genes dos nós e decidem se cooperam com as operações de escrita e leitura de um sistema de quórum. O próximo capítulo avalia a eficácia do uso do QS^2 em um sistema de quórum para MANETs com a participação de nós de má-conduta.

CAPÍTULO 5

AValiação DO ESQUEMA QS^2

Este capítulo apresenta uma avaliação do esquema QS^2 (*quorum system + quorum sensing*), empregado na exclusão de nós de má-conduta em sistemas de quórum para MANETs. A avaliação considera tanto o desempenho do QS^2 na tolerância a nós de má-conduta nas operações de um sistema de quórum quanto a eficiência na detecção desses nós. A Seção 5.1 descreve os parâmetros de simulação utilizados nos cenários de validação do QS^2 e a Seção 5.2 apresenta as métricas utilizadas para a avaliação. As Seções 5.3 e 5.4 descrevem os resultados de desempenho e eficiência do QS^2 obtidos nesses cenários e discutem os resultados alcançados.

5.1 Cenários de validação

O QS^2 foi implementado e simulado utilizando o simulador de redes *Network Simulator* (NS-2) versão 2.33. O esquema foi desenvolvido e adicionado ao código do PAN, sendo chamado de $PAN + QS^2$. Ele foi avaliado considerando a interferência de nós de má-conduta nas operações de leitura e de escrita, sendo que tais nós agem na forma de ataques de falta de cooperação, temporização e injeção de dados. Nos ataques de falta de cooperação, os nós egoístas não colaboram com as operações dos servidores do sistema de armazenamento, chamado StS. No ataque de temporização, os nós maliciosos atrasam arbitrariamente a propagação da escrita, prejudicando a execução da disseminação dos dados. Já nos ataques de injeção de dados, a cada interação com o StS, os nós maliciosos modificam os dados recebidos e injetam dados fabricados.

O ambiente de rede simulado é composto por 50 nós, sendo que metade deles compõe o StS e são escolhidos aleatoriamente no início da simulação. Os nós se comunicam por um canal sem fio, seguindo o modelo de propagação *TwoRayGround* e movimentam-se de acordo com o modelo de movimentação *Random Waypoint*, em uma área de 1000m x 1000m. O protocolo de roteamento empregado é o AODV, o raio de alcance da antena dos nós é de 250m e a velocidade máxima dos nós varia de 2m/s, 5m/s, 10m/s e 20m/s, com um tempo de pausa de 10s, 20s, 40s e 80s, similares aos parâmetros considerados na avaliação do PAN.

O quórum de leitura (Q_r) é composto por quatro servidores, incluindo o agente, e o quórum de escrita (Q_w) é formado por todos os nós que recebem a escrita de um dado. As escritas recebidas pelos nós são disseminadas a cada $T = 200\text{ms}$, e cada nó dissemina os dados para dois servidores ($fanout = 2$). Nas simulações, o intervalo de envio de escritas e leituras de cada nó é modelado seguindo a distribuição de Poisson, com $\lambda = 100$ para

as escritas e $\lambda = 36$ para as leituras, e é dado em segundos.

A quantidade esperada de escritas para cada nó do StS é $k_{env} = 15$, e é igual para todos os nós do StS. A quantidade máxima de escritas enviadas, k_{env}^{max} , é dada pela Equação (3.3), e determina-se que a probabilidade de nós confiáveis enviarem mais escritas do que o permitido seja de 5%. Desta forma, o sistema permite uma quantidade máxima de $k_{env}^{max} = 0,018$ escritas por segundo. Já a quantidade de encaminhamento esperado k_{enc} é obtida através da Equação (3.2), e a sua quantidade mínima k_{enc}^{min} para cada nó da rede é determinada pela Equação (3.4). A probabilidade de encaminhar menos dados do que k_{enc}^{min} é configurada para 5%, sendo que os nós confiáveis apresentam uma taxa de encaminhamento superior a $k_{enc}^{min} = 0,15$ pacotes por segundo.

São consideradas as seguintes porcentagens de nós de má-conduta (f): 20%, 28% e 36%. Isso corresponde a 5, 7 e 9 nós de má-conduta no StS. Esses nós agem de forma egoísta ou maliciosa durante todo o período da simulação, e em todas as vezes em que realizam uma operação nos quóruns de escrita ou de leitura. Além disso, um nó não muda a forma do seu comportamento, ou seja, um nó egoísta não apresenta comportamento malicioso e da mesma forma, um nó malicioso não muda seu comportamento para egoísta. Os resultados apresentados são as médias de 35 simulações, com um intervalo de confiança de 95%. A Tabela 5.1 resume os principais parâmetros utilizados nas simulações.

Tabela 5.1: Principais parâmetros de simulação dos cenários de validação

Parâmetros	Valor
Quantidade de nós	50
Quantidade de nós no StS	25
Tempo de vida da rede	1500 segundos
Área de movimentação	1000x1000 metros
Velocidades máximas	2m/s, 5m/s, 10m/s, 20m/s
Tempo de pausa	10s, 20s, 40s, 80s
Raio de transmissão	250 metros
<i>Fanout</i> (F)	2 servidores
Quórum de leitura	4 servidores
Intervalo de propagação (T)	200ms, 400ms, 800ms, 3000ms
Quantidade de nós de má-conduta (f)	20%, 28%, 36%
Taxa de escrita (k_{env}^{max})	0,018 escritas por segundo
Taxa de encaminhamento (k_{enc}^{min})	0,15 encaminhamentos por segundo

Os cenários considerados nas simulações correspondem à situações nas quais os usuários utilizam equipamentos sem fio, como celulares e notebooks, e movimentam-se em uma área delimitada. Esses usuários podem ser representados por pedestres ou ciclistas, que geralmente se movimentam com velocidades similares às empregadas na simulação. Um momento da simulação, retirado dos arquivos de movimentação gerados para a mobilidade dos nós, é ilustrado na Figura 5.1. Nesse cenário, os usuários são representados pelos nós, e comunicam-se diretamente com aqueles que se encontram dentro do raio de alcance da sua antena. Essa comunicação é representada por uma linha que conecta os nós cuja comunicação é feita diretamente. Informações complementares sobre os cenários

de validação e os parâmetros utilizados são descritos em [69, 70].

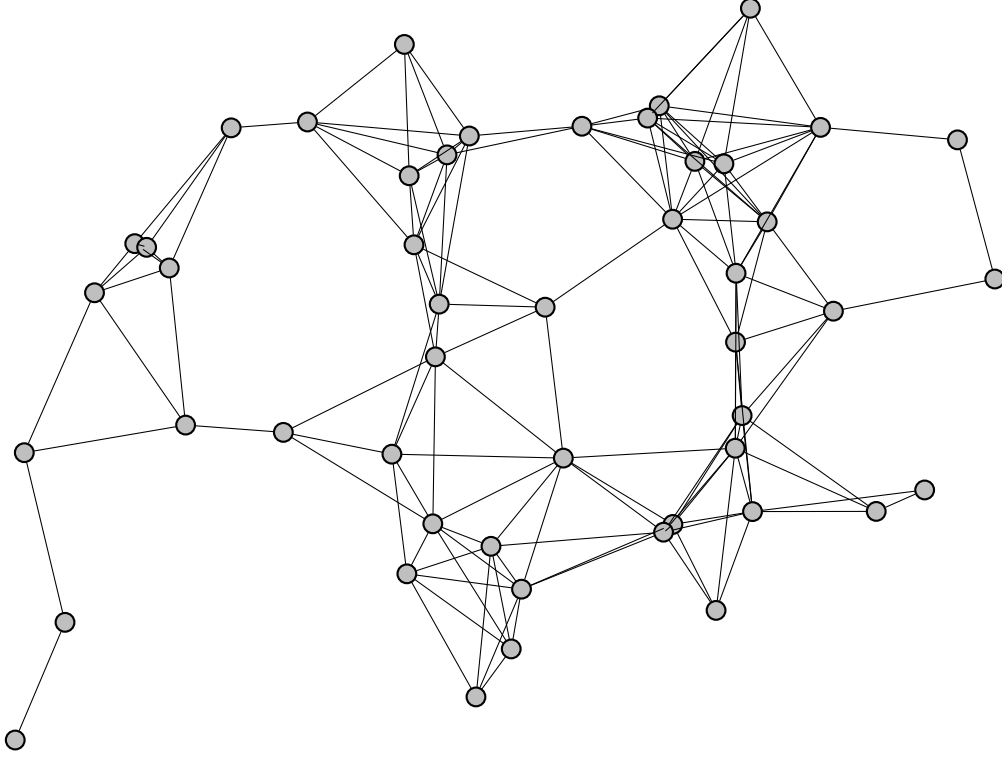


Figura 5.1: Cenário de simulação

5.2 Métricas

Foram empregadas quatro métricas para a avaliação do QS^2 diante de nós de má-conduta. A primeira delas aborda o desempenho do esquema proposto e quantifica a confiabilidade de um sistema de quórum com a participação de nós de má-conduta com o uso do QS^2 . As outras três métricas aferem a eficiência do QS^2 na identificação de nós de má-conduta.

A métrica de desempenho tem como objetivo avaliar a quantidade de escritas corretas retornada por operações de leitura no sistema de replicação. A métrica que mede o desempenho do QS^2 é o *grau de confiabilidade* (G_c), descrita na Equação (2.1) no Capítulo 2. Ela quantifica as leituras corretas retornadas pelos agentes, que representa a probabilidade de intersecção entre os quórums de escrita e de leitura formados. São consideradas corretas as leituras que obtêm um resultado correspondente à uma escrita previamente realizada no sistema ou a uma escrita ainda em progresso no momento da leitura.

As métricas de eficiência utilizadas são a *Taxa de detecção*, a *Taxa de falsos negativos* e a *Taxa de falsos positivos*. A *Taxa de detecção* (Tx_{det}) representa a porcentagem de interações de nós de má-conduta que foram detectadas pelo QS^2 . A Tx_{det} é contabilizada para os ataques de falta de cooperação e injeção de dados nas escritas. Ela é calculada de acordo com a Equação (5.1), em que A representa o conjunto de todas as interações

de nós de má-conduta e os respectivos resultados obtidos pelo QS^2 , dado na forma de $A(d, a)$, em que d é o resultado da detecção realizada pelo QS^2 e a é a verdadeira condição do nó i .

$$Tx_{det} = \frac{\sum D_i}{|A|} \forall i \in A \quad \text{onde} \quad D_i = \begin{cases} 1 & \text{se } d_i = a_i \\ 0 & \text{se } d_i \neq a_i \end{cases} \quad (5.1)$$

A terceira métrica também mede a eficiência de detecção do QS^2 e quantifica as taxas de falsos negativos obtidos na seleção de nós. A *taxa de falsos negativos* (Tx_{fn}), apresenta a quantidade de vezes em que os nós egoístas ou maliciosos foram identificados como nós confiáveis. Essa métrica é calculada pela Equação (5.2), em que A é o conjunto de todas as interações de nós de má-conduta no sistema e os respectivos resultados obtidos pelo QS^2 .

$$Tx_{fn} = |A| - Tx_{det} \quad (5.2)$$

Por fim, a quarta métrica aferida é a *taxa de falsos positivos* (Tx_{fp}), que quantifica os falsos positivos obtidos pelo QS^2 . A Tx_{fp} representa a quantidade de vezes que os nós identificaram os genes **C** e **M** ativos e consideraram um nó como malicioso ou egoísta, porém os nós eram nós confiáveis (**c** e **m**). A Tx_{fp} é calculada de acordo com a Equação (5.3), em que B representa o conjunto de interações de nós confiáveis no sistema, na forma de $B = (d, a)$, onde d representa o valor da detecção realizada pelo QS^2 e a é a condição real do nó, onde $a = 1$ representa um nó de má-conduta e $a = 0$ um nó confiável.

$$Tx_{fp} = \frac{\sum D_i}{|B|} \forall i \in B \quad \text{onde} \quad D_i = \begin{cases} 1 & \text{se } d_i = 1 \\ 0 & \text{se } d_i \neq 1 \end{cases} \quad (5.3)$$

5.3 Avaliação de desempenho

Essa seção apresenta uma avaliação da confiabilidade do PAN, um sistema de quórum probabilístico para MANETs, com o uso do QS^2 para a seleção de nós. O PAN em conjunto com o QS^2 é chamado de $PAN + QS^2$, e é submetido à interferência de nós de má-conduta em suas operações. A métrica de avaliação utilizada é o grau de confiabilidade (G_c), e os resultados do $PAN + QS^2$ são comparados aos resultados do PAN, obtidos pela avaliação apresentada no Capítulo 2. Para verificar a viabilidade do uso do QS^2 no PAN, foram simulados cenários com o $PAN + QS^2$ sem a presença de nós de má-conduta na rede. Como mostra a Figura 5.2, o $PAN + QS^2$ mantém praticamente a mesma confiabilidade que o PAN [9], com o G_c acima de 98%. O $PAN + QS^2$ até mesmo apresenta uma melhora no desempenho em velocidades mais altas, como em 10m/s e 20m/s, principalmente porque o QS^2 não seleciona para a replicação os nós que estão longe dos demais e têm menos conectividade, e dessa forma, o sistema mantém os dados

armazenados em nós altamente conectados. Isso enfatiza que o QS^2 não impõe grande custo com relação a confiabilidade alcançada e o desempenho.

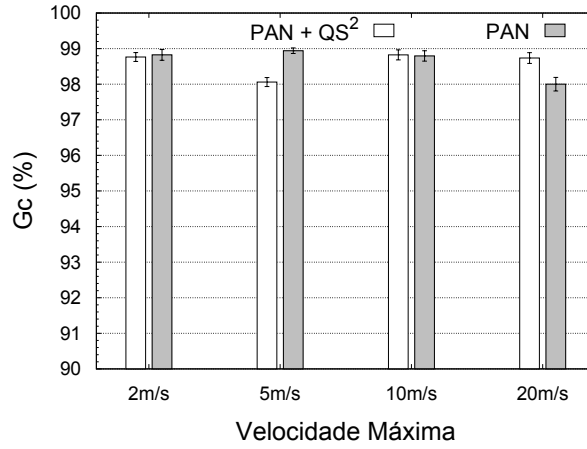


Figura 5.2: G_c do QS^2 sem ataque

5.3.1 Ataque de falta de cooperação

A Figura 5.3 apresenta os resultados referentes à métrica G_c obtidos pelo $PAN + QS^2$ com a presença de nós egoístas nas operações de escrita. Nesses cenários, os nós egoístas não colaboram com a propagação dos dados para outros nós do StS. Os resultados estão agrupados por velocidade, e são comparados com a quantidade de nós de má-conduta considerada nas simulações. São apresentados também os resultados obtidos pelo PAN sem o uso do QS^2 , e a comparação entre os resultados se dá em pontos percentuais. O $PAN + QS^2$ obteve um G_c superior em todos os cenários, representando um aumento de 4% a 9% com relação ao G_c obtido pelo PAN . O aumento do G_c é maior em velocidades mais altas, já que nessas velocidades os nós egoístas apresentam um impacto maior quando não utilizam o QS^2 para selecionar os nós.

Também nota-se que o $PAN + QS^2$ apresenta uma variação pequena no G_c em relação aos diferentes valores considerados para as velocidades, quando submetido às mesmas quantidades de nós egoístas. Isso acontece devido ao esquema de identificação e exclusão de nós de má-conduta, que não permite a participação desses nós nas operações de leitura e escrita, evitando que eles interfiram na execução das operações. O G_c obtido na presença de 5 nós egoístas na velocidade de 2m/s é de 98,3%, enquanto que na velocidade de 20m/s é de 97,1%, com a mesma quantidade de nós egoístas. A variação de aproximadamente 1% indica uma vantagem do $PAN + QS^2$, pois mostra que a velocidade dos nós na rede não apresenta uma grande interferência na execução das escritas e das leituras.

Os resultados de G_c obtidos na presença de nós egoístas nas operações de leitura no $PAN + QS^2$, apresentados na Figura 5.4, confirmam a tendência do QS^2 em manter uma variação pequena do G_c com relação a velocidades diferentes, porém com a mesma

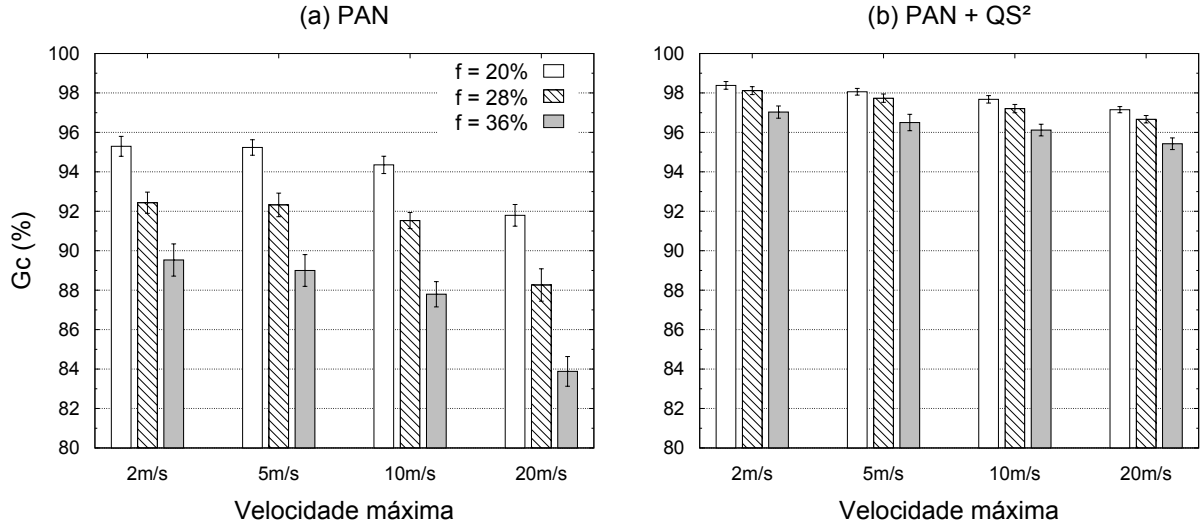


Figura 5.3: G_c com ataque de falta de cooperação na escrita

quantidade de nós egoístas. Essa variação fica próxima a 1%, assim como no cenário anterior. Além disso, os cenários com nós egoístas nas leituras apresentam um G_c superior a 98% para todas as quantidades de nós egoístas consideradas. A velocidade de 20m/s impõe um aumento mais significativo no G_c , aproximadamente 10% em cenários com 9 nós egoístas, em comparação com o PAN.

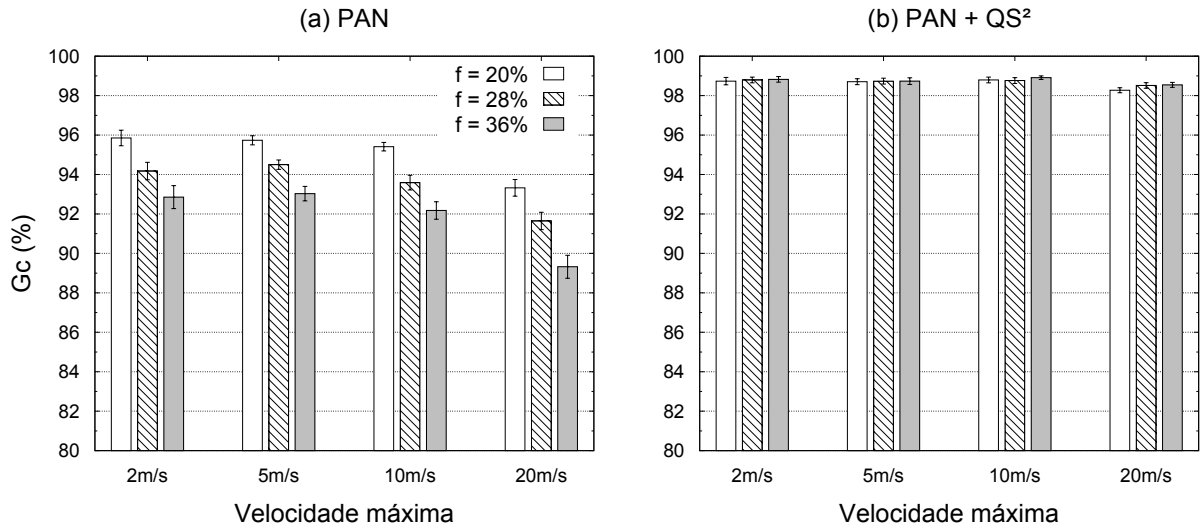


Figura 5.4: G_c com ataque de falta de cooperação na leitura

Apesar do QS² utilizar autoindutores somente para as escritas, ele permite que os nós concluam as leituras de forma correta mesmo que nós egoístas não respondam a requisição enviada pelos agentes. Isso porque o QS² garante que as escritas de nós confiáveis sejam entregues para todos os demais nós confiáveis do StS, e dessa forma, mesmo que um nó consulte um quórum de leitura e não receba nenhuma resposta, é provável que esse próprio nó já possua o valor mais atual do sistema para o dado consultado.

5.3.2 Ataque de temporização

Nos cenários em que os nós atrasam a propagação das escritas dos dados no StS o $PAN + QS^2$ não revela um aumento muito significativo nos resultados de G_c obtidos. Um dos motivos é que esse ataque não produz um grande impacto no G_c do PAN, mesmo sem a utilização do QS^2 . Dessa forma, a solução não é capaz de melhorar os resultados em grandes proporções. Outro motivo é que o QS^2 não identifica especificamente os nós que atrasam a propagação. Eles são considerados egoístas pelo sistema como consequência do seu comportamento na rede. Conforme os nós atrasam a propagação das escritas, menor é a interação deles com os demais nós do StS, e consequentemente, menor será a contagem de autoindutores para esses nós. Porém, a classificação deles como nós egoístas é demorada. A Figura 5.5 ilustra os resultados de G_c obtidos com cenários em que os nós maliciosos atrasam a propagação em $T = 400\text{ms}$ enquanto que para os nós confiáveis o $T = 200\text{ms}$.

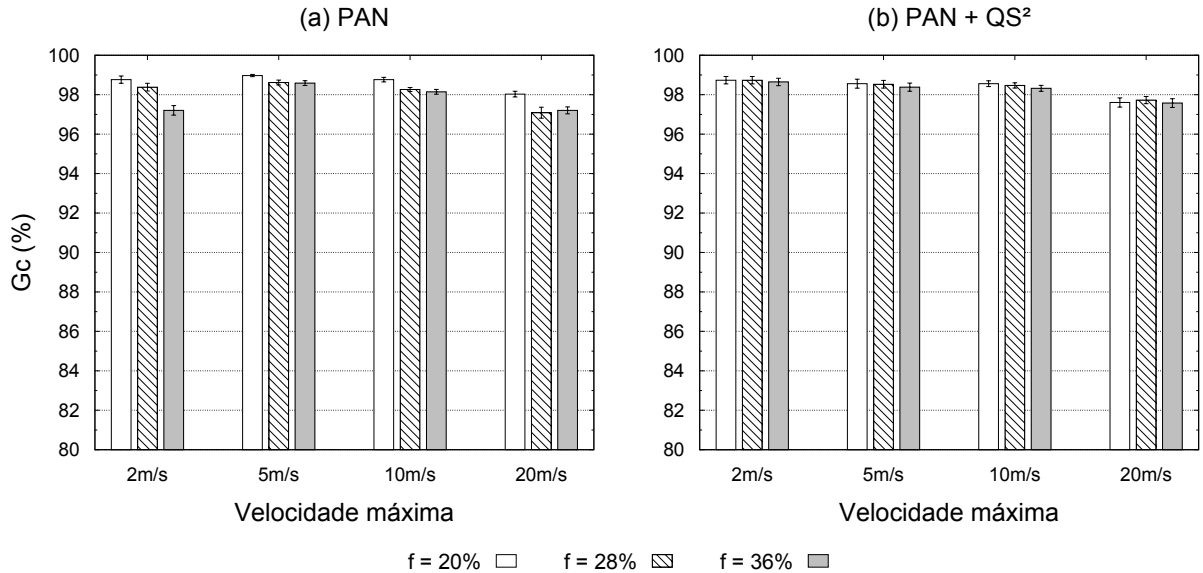


Figura 5.5: G_c com ataque de temporização com $T=400\text{s}$

Observa-se que na presença de 5 nós de má-conduta na rede, o G_c obtido pelo $PAN + QS^2$, em todas as velocidades, é ligeiramente inferior do que o obtido pelo PAN com a presença de nós de má-conduta atrasando a propagação em $T = 400\text{ms}$. Porém essa variação é pequena, sendo que a maior variação é de aproximadamente 0,42% nas velocidades de 5m/s e 20m/s. Nas demais velocidades, essa variação não passa de 0,21%. O comportamento de manter uma variação pequena entre o G_c obtido, mesmo com velocidades diferentes, pode ser um dos motivos para a queda do G_c nesses cenários. Como observado nos cenários com ataques de falta de cooperação na leitura, o $PAN + QS^2$ mantém o G_c próximo a 98%. Com o ataque de temporização, essa tendência se mantém, e isso reflete uma pequena queda do G_c em cenários em que mesmo com a presença de nós

de má-conduta, o sistema consegue concluir corretamente mais de 98% das leituras. Isso representa uma pequena queda de desempenho em alguns cenários em troca da segurança contra nós egoístas e maliciosos.

Já nos cenários com 7 e 9 nós de má-conduta, o $PAN+QS^2$ mostra um pequeno ganho com relação ao G_c obtido com o PAN. Em tais cenários, apenas a velocidade de 5m/s apresenta uma queda no G_c . Com 7 nós de má-conduta, essa queda é de 0,09% e com 9 nós de má-conduta, é de 0,2%. Já os cenários com velocidades de 2m/s representam os cenários em que o $PAN+QS^2$ tem o maior ganho, aproximadamente 0,35% na presença de 7 nós de má-conduta e 1,4% com 9 nós de má-conduta.

A Figura 5.6 ilustra os resultados obtidos em cenários com nós atrasando a propagação das escritas em $T = 800ms$.

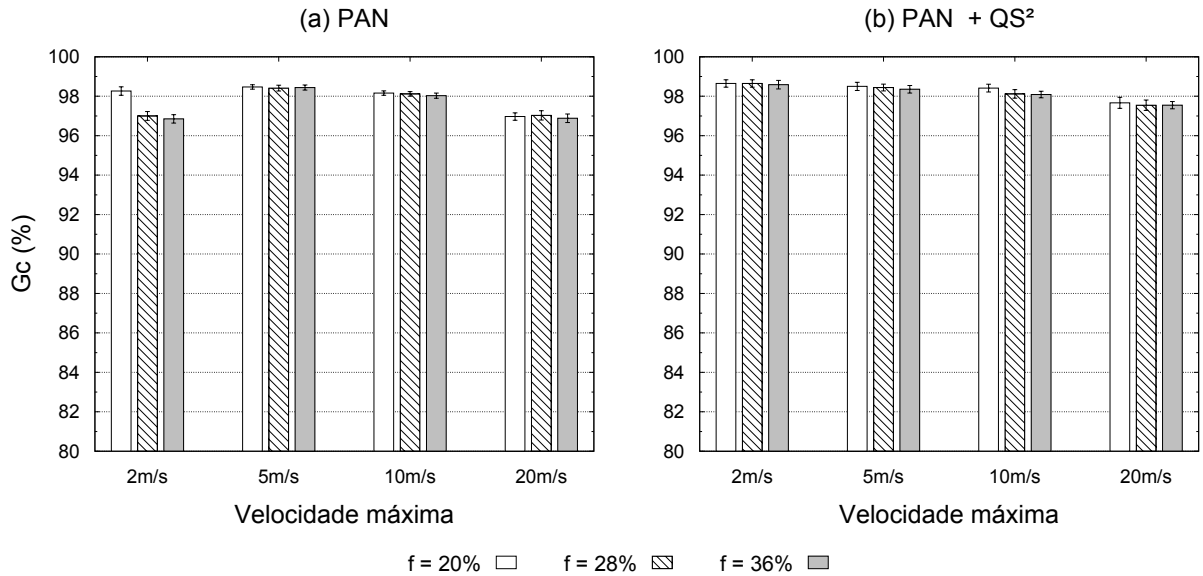


Figura 5.6: G_c com ataque de temporização com $T=800s$

Os cenários com nós de má-conduta atrasando as escritas em $T = 800ms$ e $T = 3000ms$ apresentam um ganho mais acentuado do G_c . Nesses cenários, o ganho do $PAN + QS^2$ chega a 0,5% com 5 nós de má-conduta e velocidade de 20m/s, e de aproximadamente 1,8% em cenários com 9 nós de má-conduta e velocidade de 2m/s. Também é possível observar uma pequena queda em cenários com velocidade de 10m/s, porém o declínio do G_c é de somente 0,09%.

Já nos cenários com nós atrasando a propagação de dados em 3 segundos, o ganho com a utilização do QS^2 é mais acentuado, conforme apresentado na Figura 5.7. Apesar de uma queda de aproximadamente 0,04% nas velocidades de 5m/s em cenários com 7 e 9 nós de má-conduta, o G_c com a mesma quantidade de nós de má-conduta, porém com velocidade de 2m/s, tem um aumento de aproximadamente 2%, a maior porcentagem dentre os cenários com nós que atrasam a propagação dos dados.

Nesse tipo de ataque, o QS^2 não se mostra eficaz para cenários em que os nós de má-

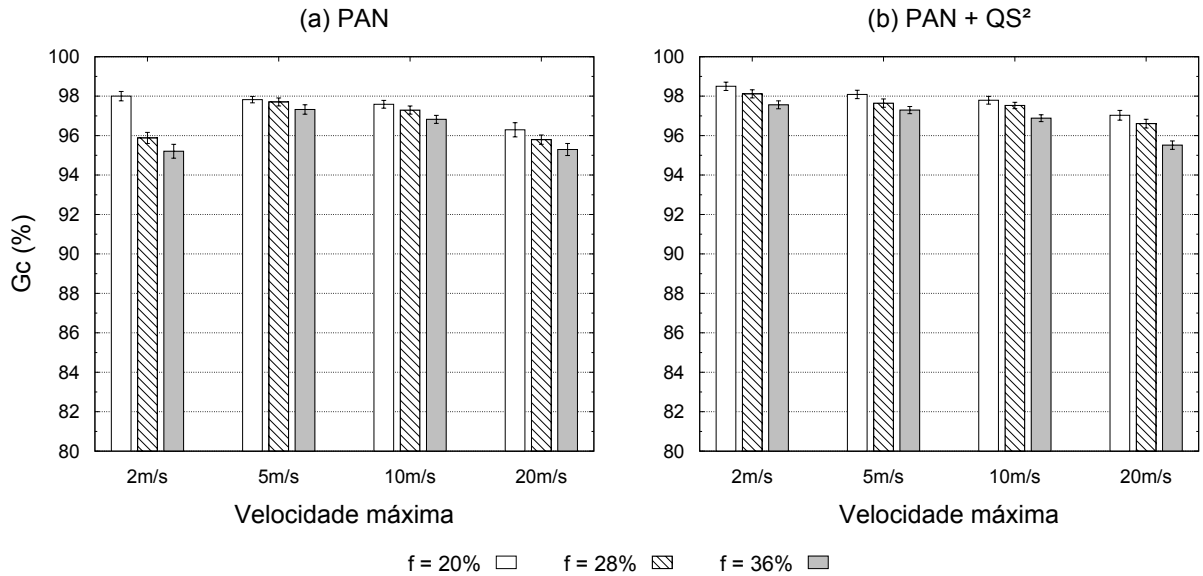


Figura 5.7: G_c com ataque de temporização com $T=3000s$

conduta não conseguem comprometer as operações em grande escala. Isso porque o QS^2 apresenta uma tendência em nivelar o G_c para as diferentes velocidades, ocasionando uma troca entre o desempenho e a segurança contra ataques de falta de cooperação e injeção de dados, abordados a seguir. Será observado que o ganho do sistema diante de ataques de injeção de dados compensa a pequena perda do G_c apresentado em alguns cenários com ataques de temporização.

5.3.3 Ataque de injeção de dados

O uso do QS^2 diante de ataques de injeção de dados apresenta um ganho significativo para o PAN em comparação com os resultados obtidos sem a solução. Esse ataque representa a maior vulnerabilidade do PAN, conforme a avaliação mostrada no Capítulo 2. Esses cenários também evidenciam a tendência do QS^2 em manter o G_c em uma mesma taxa em diferentes velocidades. A Figura 5.8 ilustra os resultados de G_c obtidos pelo $PAN + QS^2$ em cenários com nós maliciosos, que injetam dados falsos nas operações de escrita. Nesses cenários, o $PAN + QS^2$ apresenta um aumento de até 87% no G_c , como observado em cenários com 5 nós maliciosos movimentando-se a 2m/s.

Conforme o número de nós maliciosos aumenta, o G_c obtido pelo $PAN + QS^2$ tem um pequeno declínio. Com 5 nós maliciosos, o G_c é em média 92%, enquanto que com 7 nós maliciosos é 86% e com 9 nós maliciosos é 80%. Contudo, em nenhum cenário o $PAN + QS^2$ apresenta um G_c inferior a 80%. Observa-se também que as características da rede fazem com que o $PAN + QS^2$ obtenha níveis mais altos de G_c com velocidades maiores. Esse comportamento também é observado no PAN diante de ataques, e acontece porque os nós maliciosos perdem sua eficácia em velocidades maiores, devido à dificuldade

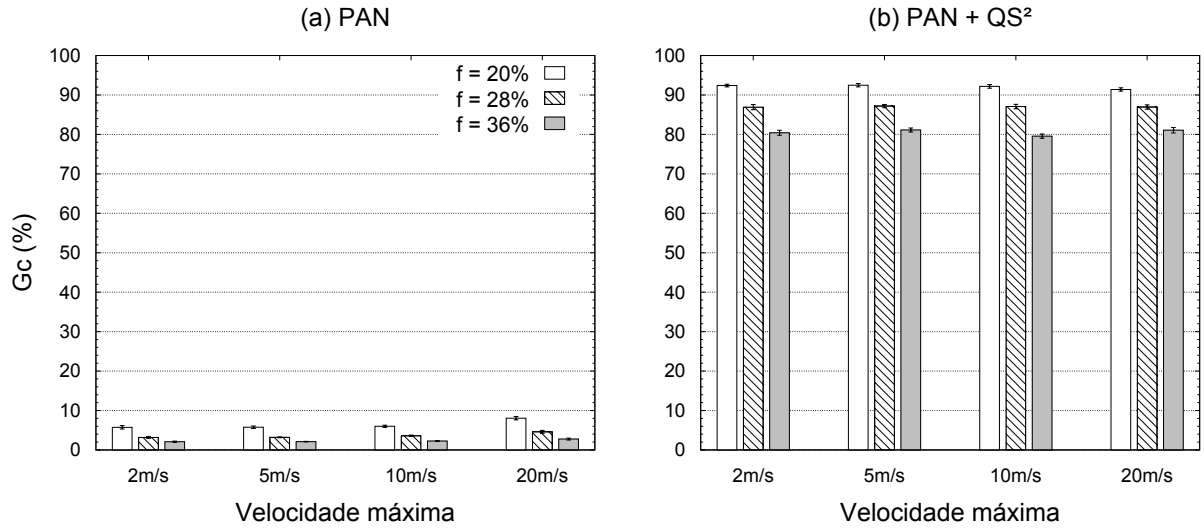


Figura 5.8: G_c com ataque de injeção de dados na escrita

na entrega de pacotes em geral, inclusive os pacotes falsos injetados pelos nós maliciosos.

Já nos cenários com nós maliciosos interagindo na leitura dos dados, o $PAN + QS^2$ obtém uma vantagem ainda maior com a junção dos esquemas de seleção e a comparação dos resultados obtidos do quórum de leitura em operações de leitura. Nesses cenários, apresentados na Figura 5.9, o aumento do G_c é de até 68%, como observado em cenários com 9 nós maliciosos e velocidade de 2m/s. O G_c obtido pelo $PAN + QS^2$ mantém uma média de 97% em todos os cenários, independentemente da quantidade de nós maliciosos.

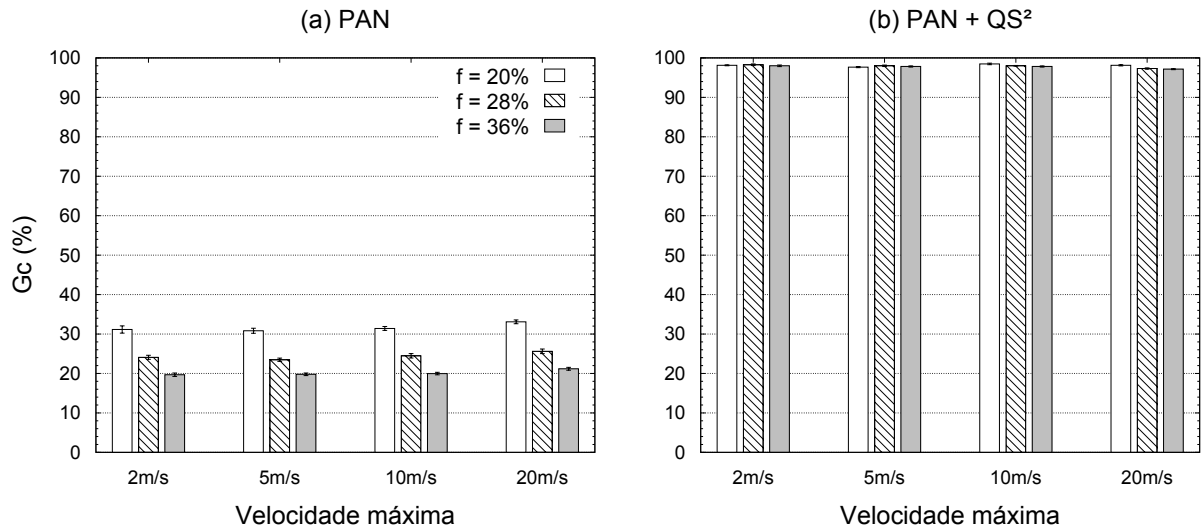


Figura 5.9: G_c com ataque de injeção de dados na leitura

Nos cenários com ataques na operação de leitura, o $PAN + QS^2$ além de identificar os nós maliciosos, também evita que os dados eventualmente enviados por eles sejam escolhidos como resposta de uma operação de leitura. Isso porque o QS^2 exige que uma resposta correta seja uma que mais de um nó respondeu com o mesmo valor, o que não é

o caso de nós maliciosos na rede, que injetam dados falsos independentemente.

5.3.4 Nós egoístas e maliciosos

Para testar a eficiência do $PAN + QS^2$ diante de vários ataques, configurou-se um cenário com a presença de nós de má-conduta no StS, que podem iniciar ataques de falta de cooperação, temporização e injeção de dados em um mesmo cenário. Para isso, foram configurados cenários com 5, 10 e 15 nós de má-conduta. Nos cenários com 5 nós de má-conduta, cada nó inicia um dos seguintes ataques: falta de cooperação na escrita, falta de cooperação na leitura, temporização com atraso de 3 segundos, injeção de dados na escrita e injeção de dados na leitura. Os cenários com a presença de 10 nós de má-conduta contêm 2 nós iniciando cada um dos ataques anteriores, e os cenários com 15 nós de má-conduta possuem 3 nós iniciando cada um desses ataques. Foram considerados cenários com velocidades de 0m/s a 20m/s, com o tempo de pausa conforme a Tabela 5.2. Os demais parâmetros da simulação são os mesmos da avaliação do $PAN + QS^2$, em que $k_{env}^{max} = 0,018$ e $k_{enc}^{min} = 0,15$.

Tabela 5.2: Relação de velocidade máxima e tempo de pausa

Velocidade	Pausa
0m/s - 4m/s	10s
5m/s - 9m/s	20s
10m/s - 14m/s	40s
15m/s - 20m/s	80s

A Figura 5.10 apresenta os resultados obtidos em cenários com vários nós de má-conduta iniciando diversos ataques na rede. Os resultados estão organizados por quantidade de nós de má-conduta. É possível observar que conforme a quantidade de nós de má-conduta aumenta o G_c diminui, porém enquanto a quantidade de nós de má-conduta é a mesma, a variação do G_c de acordo com a velocidade é pequena, mostrando que a solução tende a manter um mesmo nível de leituras corretamente concluídas, independente da velocidade. Essa variação, em todos os cenários de diferentes quantidades de nós de má-conduta, é de aproximadamente 1%. Esse comportamento representa uma vantagem ao sistema, já que os nós das MANETs podem variar a velocidade e ainda assim manter um bom percentual de leituras corretas, mesmo diante de um StS com mais de 50% dos nós comprometidos.

Também foi possível observar que a cada incremento de 5 nós de má-conduta na rede, o percentual de queda em relação ao anterior foi menor do que 5%. Com a presença de 5 nós de má-conduta no StS, nos cenários com velocidade igual a 8m/s, o G_c foi igual a 98,4%, enquanto que o G_c para a mesma velocidade nos cenários com 10 nós de má-conduta foi de 95,9% e com 15 nós de má-conduta foi de 91,9%. É interessante notar que mesmo com metade do StS comprometido com nós de má-conduta, o $PAN + QS^2$ obteve

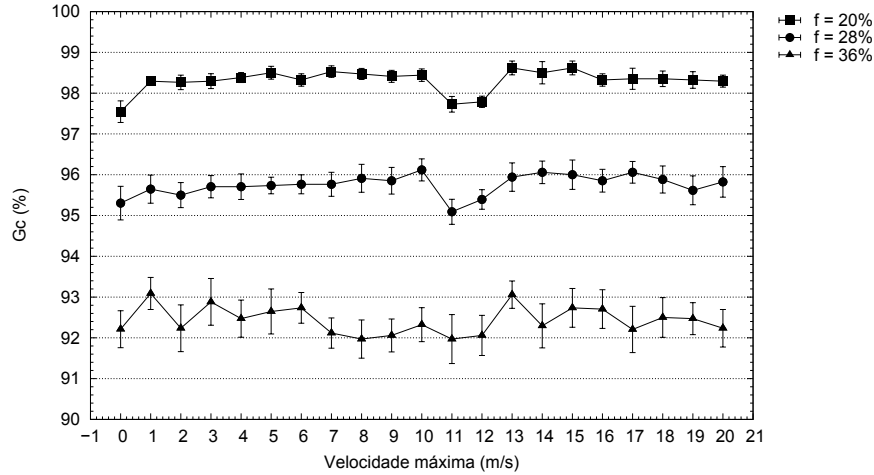


Figura 5.10: G_c em uma MANET com nós egoístas e maliciosos

um G_c acima de 92% para todos os cenários simulados.

5.4 Avaliação de eficiência

Essa seção apresenta a avaliação da eficiência do QS^2 na detecção de nós egoístas e maliciosos nas operações de um sistema de quórum probabilístico para MANETs. As métricas que quantificam a eficiência são a taxa de detecção (Tx_{det}), a taxa de falsos negativos (Tx_{fn}) e a taxa de falsos positivos (Tx_{fp}) obtidos pelo QS^2 .

5.4.1 Taxa de detecção

A Figura 5.11 apresenta a taxa de detecção de nós de má-conduta obtida pelo QS^2 . A taxa de detecção representa a quantidade de nós corretamente identificados como nós egoístas ou maliciosos, em todas as interações deles com o sistema. Os resultados são apresentados para os ataques de falta de cooperação e injeção de dados, que possuem nós egoístas e maliciosos que são detectados pelo QS^2 . Nos ataques de falta de cooperação, apresentado na Figura 5.11(a), o QS^2 apresenta uma taxa de detecção superior a 98,5%. Isso ocorre porque o QS^2 identifica corretamente os nós egoístas, que uma vez identificados, só serão considerados confiáveis novamente se cooperarem com os demais nós. Se o comportamento egoísta se mantém, os nós são excluídos do sistema e não são mais consultados. Esse comportamento é observado em todas as velocidades, e é independente da quantidade de nós de má-conduta presentes no ambiente.

Já no ataque de injeção de dados, ilustrado na Figura 5.11(b), a taxa de detecção de nós maliciosos apresenta uma variação mais acentuada do que na detecção de ataques de falta de cooperação, sendo que a média de detecção é próxima de 80%. Essa variação ocorre tanto entre as velocidades com a mesma quantidade de nós maliciosos como entre os cenários com quantidades diferentes de nós maliciosos. Nos resultados obtidos nos

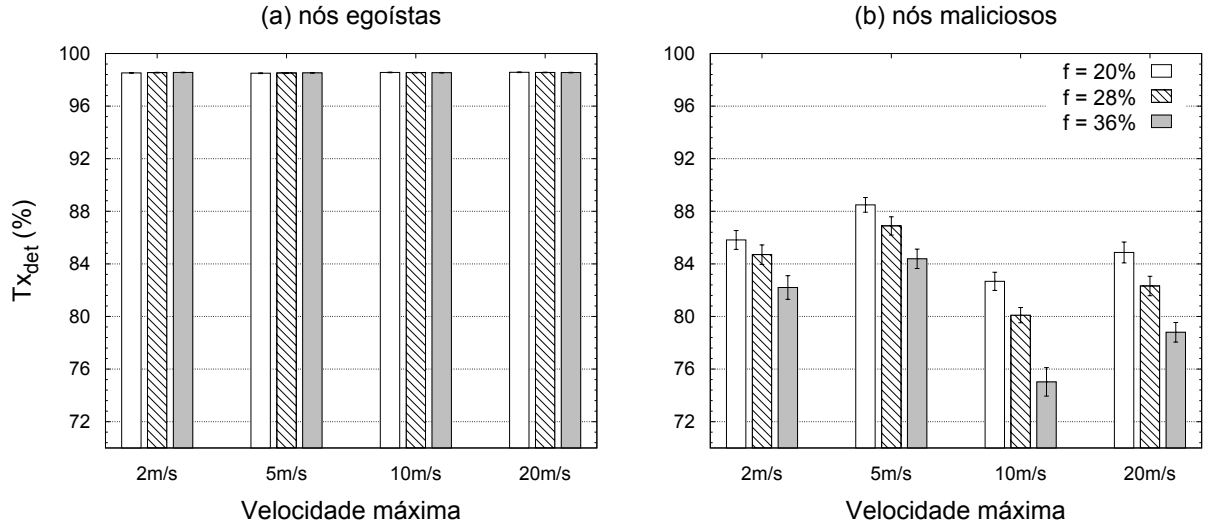


Figura 5.11: Tx_{det} de nós egoístas e maliciosos

cenários com 5 e 7 nós maliciosos, há uma variação de aproximadamente 6% entre a eficácia da detecção nas diferentes velocidades, e nos cenários com 9 nós maliciosos a variação é de aproximadamente 9%. Isso ocorre porque o QS^2 identifica os nós maliciosos pelo comportamento em um determinado intervalo de tempo, não interagindo com os nós classificados como maliciosos. Com o passar do tempo, os nós maliciosos não são mais contatados, diminuindo a interação deles com o sistema, o que resulta na normalização do nível de autoindutores nos demais nós do sistema. Isso leva os nós confiáveis a considerá-los confiáveis e a interagir novamente com eles, o que diminui a eficácia devido aos falsos negativos.

5.4.2 Taxa de falsos negativos

Nos ataques de falta de cooperação, o QS^2 apresenta uma pequena taxa de falsos negativos, como mostra a Figura 5.12(a). Essa taxa é inferior a 2%, e mostra que poucos nós egoístas não foram detectados quando selecionados. Isso pode acontecer devido a autonomia na detecção, que permite que os nós contem individualmente os autoindutores dos demais nós. Dessa forma, alguns nós podem demorar a identificar determinados nós como egoístas, e até então selecioná-los para as operações de leitura e de escrita nos sistemas de quórum.

A taxa de falsos negativos no ataque de injeção de dados, apresentada pela Figura 5.12(b), é mais elevada em comparação a Tx_{fn} obtidas com o ataque de falta de cooperação. Nesse caso, a média de falsos negativos é de 20%, sendo maior em cenários com mais nós de má-conduta participando na rede. Esse aumento de falsos negativos no ataque de injeção de dados acontece porque os nós maliciosos, quando identificados, deixam de participar das operações dos quóruns. Isso motiva os nós a não enviar dados

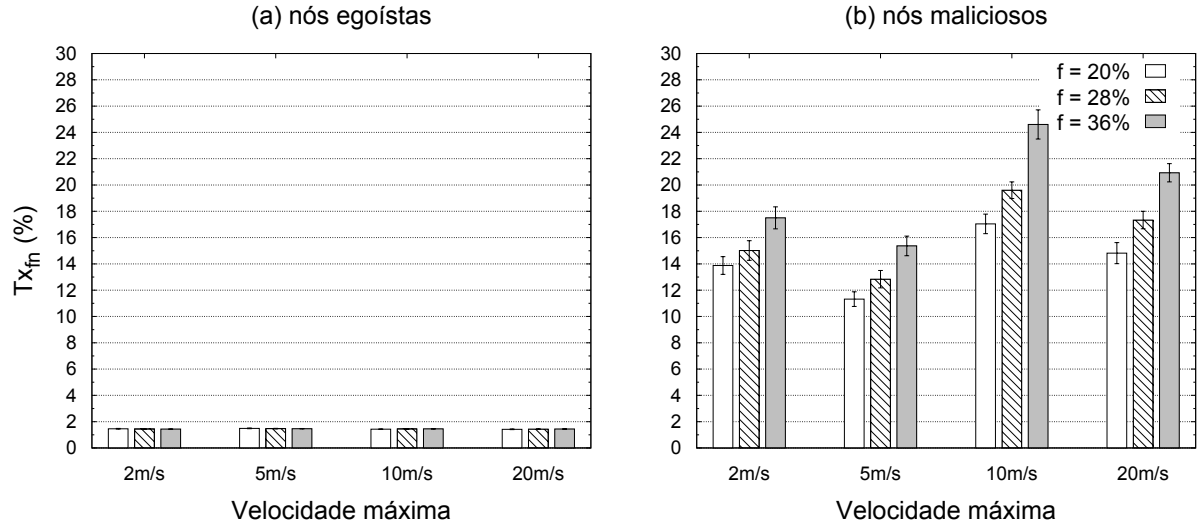


Figura 5.12: Tx_{fn} na detecção de nós egoístas e maliciosos

maliciosos na rede, e implica na normalização da contagem de autoindutores para esses nós. Com o passar do tempo, os nós voltam a identificá-los como maliciosos. Portanto, apesar da taxa de falsos negativos ser maior que nos ataques de falta de cooperação, o QS^2 identifica e exclui os nós maliciosos com a mesma eficiência.

5.4.3 Taxa de falsos positivos

O QS^2 obteve uma baixa taxa de falsos positivos, conforme mostra a Figura 5.13. Nos cenários com ataques de falta de cooperação, ilustrado na Figura 5.13(a), a taxa de falsos positivos não passa de 2%, enquanto que nos cenários com ataques de injeção de dados, apresentados na Figura 5.13(b), a Tx_{fp} é menor que 1%. Isso indica que o QS^2 identifica poucos nós confiáveis como egoístas ou maliciosos. Essa detecção equivocada pode acontecer se um nó está muito distante na rede e apresenta dificuldade em interagir com o restante da rede, ou se um nó faz muitas escritas contínuas para o mesmo grupo de nós. Deste modo, momentaneamente eles são considerados nós de má-conduta, porém conforme ocorre a movimentação e a interação dos nós, eventualmente eles são identificados como nós confiáveis.

Em ambos os ataques a Tx_{fp} tem uma variação pequena, independente da velocidade ou da quantidade de nós de má-conduta. No ataque de falta de cooperação com 5 nós de má-conduta e velocidade de 2m/s, a Tx_{fp} é de 1,24% e com 9 nós de má-conduta e velocidade de 2m/s o Tx_{fp} é de 1,21%. Já nos ataques de injeção de dados, o Tx_{fp} com 5 nós de má-conduta e velocidade de 2m/s é de 0,7%, e com 9 nós movimentando-se na mesma velocidade o Tx_{fp} é de 0,6%.

A Figura 5.14 ilustra a frequência com que os nós do StS são detectados como nós de má-conduta. Essa figura representa um cenário com 5 nós maliciosos, sendo que os nós

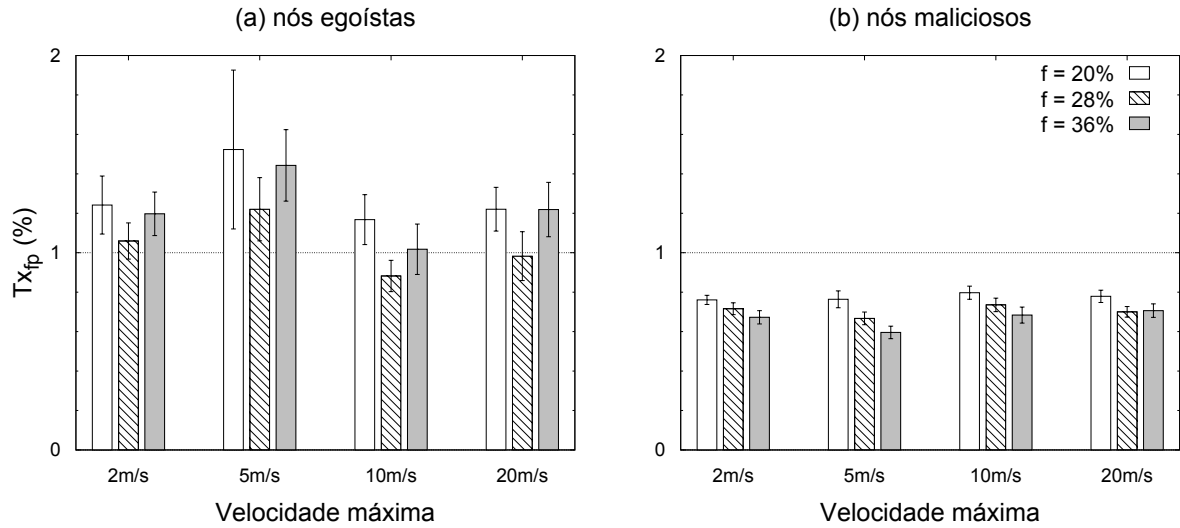


Figura 5.13: Tx_{fp} na detecção de nós egoístas e maliciosos

3, 16, 18, 19 e 22 são efetivamente os nós maliciosos. É possível observar que o QS^2 identifica os nós maliciosos com muito mais frequência do que os demais nós, e que os nós que não são maliciosos podem ser eventualmente identificados como nós de má-conduta, porém em uma escala muito menor se comparada com a detecção dos nós de má-conduta.

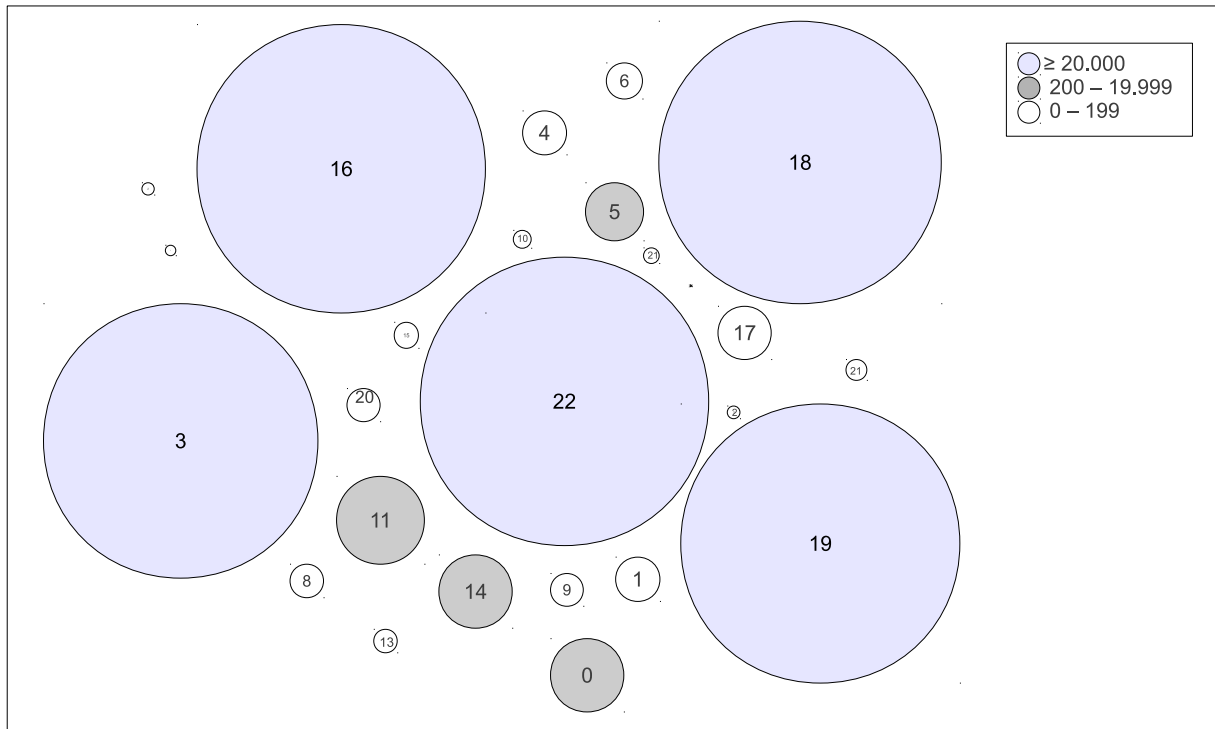


Figura 5.14: Infográfico da frequência de detecção de nós maliciosos pelo QS^2

Uma síntese do ganho na confiabilidade do PAN ao empregar o QS^2 para a seleção de nós é apresentada na Tabela 5.3. Os resultados são comparados com os obtidos na análise do PAN diante de nós egoístas e maliciosos. Com a presença de nós egoístas, o ganho na

confiabilidade nos ataques nas operações de leitura é de 3%, e nos ataques nas operações de escrita é de 4%. O ganho na confiabilidade diante do ataque de temporização é de 3% em média. Já o ataque de manipulação de dados, que representou a maior vulnerabilidade do PAN conforme apresentado no Capítulo 2, tem um ganho superior aos outros ataques. Na presença de nós maliciosos nas operações de leitura, o ganho na confiabilidade é de 67%, e nas escritas o ganho é em média de 83%.

Tabela 5.3: Síntese do ganho na confiabilidade com o uso do QS^2

Ataques	Ganho nas leituras	Ganho nas escritas
Falta de cooperação	3%	4%
Temporização	-	3%
Injeção de dados	67%	83%

A Tabela 5.4 resume os resultados das métricas de eficiência do QS^2 . A taxa de detecção dos nós, Tx_{det} , é de aproximadamente 98% para os nós egoístas e de 86% para os nós maliciosos. A taxa de falsos negativos Tx_{fn} na detecção de nós egoístas é de 1,5% e para os nós maliciosos é de aproximadamente 18%. Já as taxas de falsos positivos Tx_{fp} na detecção de nós egoístas é de 1,2% e para nós maliciosos é de 0,8%.

Tabela 5.4: Síntese da eficiência do QS^2

Métrica de eficiência	Nós egoístas	Nós maliciosos
Taxa de detecção (Tx_{det})	98%	86%
Taxa de falsos negativos (Tx_{fn})	1,5%	18%
Taxa de falsos positivos (Tx_{fp})	1,2%	0,8%

5.5 Resumo

Esse capítulo apresentou uma avaliação do uso do QS^2 na seleção de nós em operações de leitura e de escrita em um sistema de quórum para MANETs diante de cenários com nós de má-conduta. Os resultados mostraram que o QS^2 tem um bom desempenho na tolerância a ataques de falta de cooperação e injeção de dados, sendo que o ganho na confiabilidade dos sistemas de quórum com o uso do QS^2 em relação ao PAN foi mais significativo em ataques de injeção de dados. Além disso, o QS^2 foi eficiente na detecção de nós egoístas e maliciosos, apresentando uma baixa taxa de falsos positivos e negativos.

CAPÍTULO 6

SERVIÇOS DE OPERAÇÃO DE REDE CONFIÁVEIS

Este capítulo apresenta a aplicação e a avaliação do esquema QS^2 no provimento de serviços confiáveis em cenários realísticos de MANETs, através da replicação de dados utilizando um sistema de quórum, o PAN. Dois cenários de MANETs são considerados, onde o primeiro trata da distribuição de informações no centro de uma cidade e o segundo representa a distribuição de informações de tráfego em linhas de ônibus. A avaliação considera tanto o desempenho do QS^2 na tolerância a nós de má-conduta nas operações de um sistema de quórum quanto a eficiência na detecção desses nós. A Seção 6.1 descreve os serviços de operação de rede considerados nos cenários. A Seção 6.2 apresenta os parâmetros de simulação e os resultados de desempenho e eficiência do QS^2 no apoio aos sistemas de quórum em MANETs no centro de uma cidade. A Seção 6.3 introduz os parâmetros empregados para a simulação de sistemas de quórum utilizando o QS^2 em MANETs implantadas em linhas de ônibus e apresenta os resultados alcançados.

6.1 Serviços de operação de rede

As características inerentes das MANETs trazem a necessidade de monitoramento e configuração de vários aspectos da rede. Os serviços de operação são primordiais para o apoio ao correto funcionamento dessas redes, e precisam ser confiáveis e tolerantes a nós de má-conduta. Nesta seção, enfatiza-se a importância de gerenciar os dados dos diferentes serviços de operação de rede, como localização de recursos e informação de mobilidade, a fim de prover confiabilidade e disponibilidade desses dados. A falta de informação ou informações desatualizadas influenciam na operação dos nós, dos serviços e das aplicações, comprometendo o desempenho da rede.

A confiabilidade assegurada pelo QS^2 atende aos requisitos de aplicações cuja garantia de disponibilidade é mais relevante que o custo de lidar com eventuais inconsistências dos dados replicados. Exemplos dessas aplicações são o monitoramento de ambientes, a distribuição de informações de alerta ou tráfego para veículos e a distribuição de informações para pedestres. Desta forma, empregou-se o QS^2 no suporte ao sistema de quórum PAN em dois cenários realísticos de MANETs. No primeiro cenário, considerou-se o uso do QS^2 em MANETs criadas no centro de uma cidade com o objetivo de disseminar informações sobre o comércio local entre os usuários da rede. No segundo cenário, o QS^2 é utilizado no apoio do sistema de quórum aplicado em MANETs em linhas de ônibus, que se destinam a disseminar informações de tráfego e informações dos horários de ônibus. Os parâmetros e os detalhes de configuração dos cenários utilizados são descritos nas seções seguintes.

6.2 Ambiente urbano - centro de uma cidade

O cenário urbano utilizado corresponde ao ambiente do centro de uma cidade, onde pedestres e ciclistas se movimentam em direção a pontos de interesse, como *shoppings* e teatros. Estes cenários têm como base o cenário descrito em [71], em que considera-se um ambiente que representa, em geral, os centros urbanos de cidades europeias. A ideia dos autores é criar uma rede cuja função seja a disseminação de mensagens entre os dispositivos móveis dos transeuntes e os dispositivos localizados em lojas, restaurantes e demais pontos de interesse. Considera-se que tais mensagens enviadas são informativas, como promoções de lojas e cardápio dos restaurantes, assim como mensagens de utilidade pública e informações sobre condições das vias e do transporte público. Para receber as mensagens, os usuários devem se cadastrar e informar o interesse no recebimento de tais mensagens. Além disso, o usuário receberá as mensagens enquanto estiver no raio de alcance dos demais dispositivos da rede. Nesse sentido, é necessário que os serviços de operação da rede sejam robustos, garantindo a eficácia da rede e evitando que dispositivos maliciosos comprometam o seu funcionamento e, conseqüentemente, o funcionamento da aplicação. Dessa forma, empregou-se o QS^2 para o apoio aos sistemas de quórum, que suportam os serviços de operação de rede.

O cenário é ilustrado na Figura 6.1 e consiste de pontos de interesse de usuários (vértices), como igrejas, universidades e terminais de ônibus, interligados por diversas ruas (arestas). Os usuários se movimentam pelas ruas em direção a algum ponto de interesse, seguindo o padrão de mobilidade baseada em grafos (*graph walk*) [72]. Esse padrão de movimentação é mais realístico do que o empregado no cenário de avaliação anterior, em que os nós se movimentam randomicamente (*random waypoint*), sem seguir um caminho pré-definido.

Foram considerados cenários compostos por 50, 100 e 150 nós, sendo possível avaliar o impacto da densidade da rede no funcionamento do QS^2 . Os nós movimentam-se com velocidades entre 3 e 5km/h, normais para um pedestre, em uma área de 2462 x 1733 metros. Essa área compreende 75 pontos em comum, interligados por 116 ruas. Os nós escolhem randomicamente um dos pontos de interesse e movimentam-se até ele por meio das ruas, permanecendo no ponto escolhido entre 12 e 20 minutos. O tempo de pausa utilizado representa a parada de pedestres em terminais de ônibus ou *shoppings*. Após o término do tempo de pausa, o nó escolhe um novo destino e movimenta-se até ele. O AODV é empregado como protocolo de roteamento nas simulações. Os resultados apresentados são a média de 35 simulações de 1500 segundos, com um intervalo de confiança de 95%.

Alguns parâmetros referentes ao sistema de quórum PAN foram adaptados por conta do aumento dos nós na rede e do ambiente, de forma que os quórums não sejam prejudicados pela perda de mensagens ou pelo atraso na propagação das mensagens. O quórum

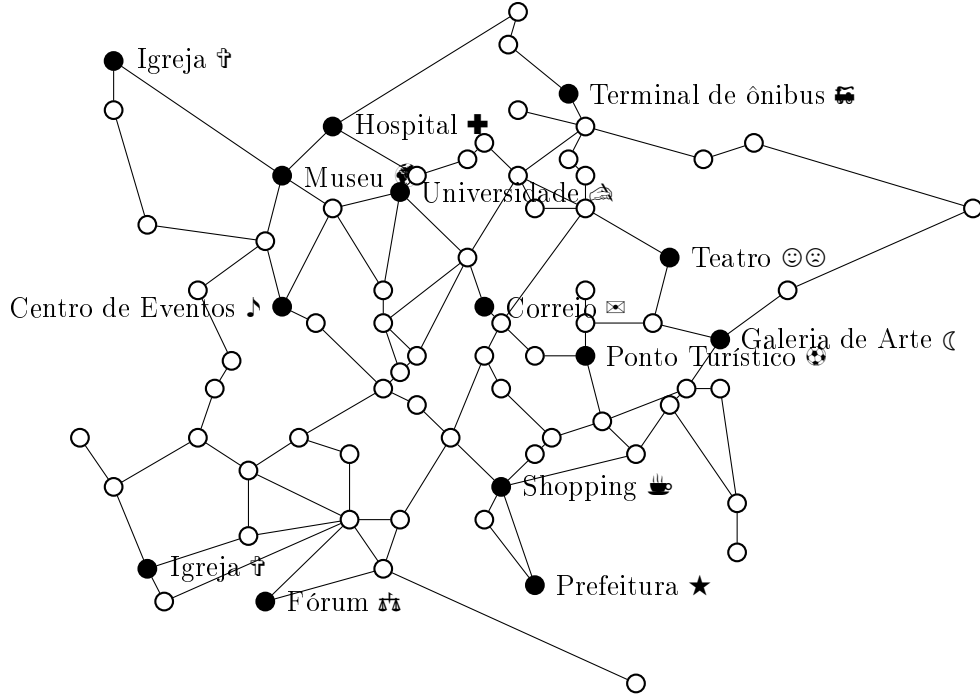


Figura 6.1: Cenário de simulação do centro de uma cidade [67]

de leitura (Q_r) é composto por cinco servidores, incluindo o agente, e o quórum de escrita (Q_w) é formado por todos os nós que recebem a escrita de um dado, sendo que cada nó dissemina os dados para quatro servidores ($F = 4$), ao invés de dois servidores como nas simulações anteriores. Os demais parâmetros referentes ao sistema de quórum e ao QS^2 são os mesmos utilizados nas simulações anteriores: o StS é composto por 25 nós, escolhidos randomicamente. As escritas recebidas pelos nós são disseminadas a cada $T = 200\text{ms}$. Nas simulações, o intervalo de envio de escritas e leituras de cada nó é modelado seguindo a distribuição de Poisson, conforme utilizado em [9], com $\lambda = 100$ para as escritas e $\lambda = 36$ para as leituras, e é dado em segundos. A taxa máxima de escritas é igual a $k_{env}^{max} = 0,018$ escritas por segundo, e a taxa de encaminhamento deve ser superior a $k_{enc}^{min} = 0,15$ pacotes por segundo. Um resumo dos principais parâmetros empregados nas simulações dos cenários urbanos é apresentado na Tabela 6.1.

Os ataques foram separados em dois conjuntos: um cenário com ataques de injeção de dados nas operações de escrita e um cenário com os ataques de falta de cooperação, temporização e injeção de dados agindo em conjunto. Nos ataques de injeção de dados foram considerados cenários com f igual a 5, 7 e 9 nós de má-conduta, o que representa 20%, 28% e 36% dos nós pertencentes ao StS. Já nos cenários com todos os ataques, foram simulados cenários com f igual a 5, 10 e 15 nós de má-conduta, representando 20%, 40% e 60% dos nós do StS. Os ataques nesse cenário foram os de falta de cooperação nas leituras e nas escritas, temporização ($T=3000\text{ms}$) e injeção de dados na leitura e na escrita, sendo

Tabela 6.1: Principais parâmetros de simulação dos cenários urbanos

Parâmetros	Valor
Quantidade de nós	50, 100 150
Quantidade de nós no StS	25
Tempo de vida da rede	1500 segundos
Velocidades máximas	3 a 5km/h
Tempo de pausa	12 a 20 minutos
Raio de transmissão	250 metros
<i>Fanout</i> (F)	4 servidores
Quórum de leitura	5 servidores
Intervalo de propagação (T)	200ms e 3000ms
Quantidade de nós de má-conduta (f)	20%, 28%, 36%
Taxa de escrita ($k_{env^{max}}$)	0,018 escritas por segundo
Taxa de encaminhamento ($k_{enc^{min}}$)	0,15 encaminhamentos por segundo

que cada ataque é desempenhado por 20% do total de nós de má-conduta presente em cada cenário.

As métricas utilizadas para avaliar o cenário urbano foram o *grau de confiabilidade* (G_c) e a *taxa de detecção* (Tx_{det}), descritas nas Equações 5.1 e 5.2, respectivamente. Além dessas, outras duas métricas foram introduzidas para quantificar os dados falsos e os dados desatualizados no sistema. A métrica Tx_{mis} , descrita na Equação (6.1), quantifica as leituras que retornaram dados escritos por nós maliciosos, no ataque de injeção de dados, em que C_w representa a quantidade de leituras que retornaram dados falsos e R representa o total de leituras realizadas no sistema. A métrica Tx_{out} , calculada de acordo com a Equação (6.2), quantifica as leituras desatualizadas retornadas pelos nós do StS. Considera-se que um dado esteja desatualizado se o seu valor não corresponder com a última escrita realizada no sistema, ou ainda com uma eventual escrita ainda em progresso. A Tx_{out} é dada pela subtração da quantidade de leituras falsas do montante de leituras corretas realizadas pelo sistema.

$$Tx_{mis} = \frac{\sum C_w}{|R|} \quad (6.1)$$

$$Tx_{out} = G_c - Tx_{mis} \quad (6.2)$$

6.2.1 Grau de confiabilidade

Os resultados obtidos pela simulação das MANETs criadas para a distribuição de informações no centro de uma cidade, sem o uso do QS^2 , são apresentados na Figura 6.2. A confiabilidade dos dados nesses cenários, tanto diante de ataques de injeção de dados (Figura 6.2(a)) quanto diante de todos os ataques (Figura 6.2(b)), é inferior a 10%. Os resultados são condizentes com os estudos prévios, e evidenciam que diante de ataques de injeção de dados, sem o emprego de uma solução de segurança, não é possível obter uma confiabilidade dos dados acima de 10%.

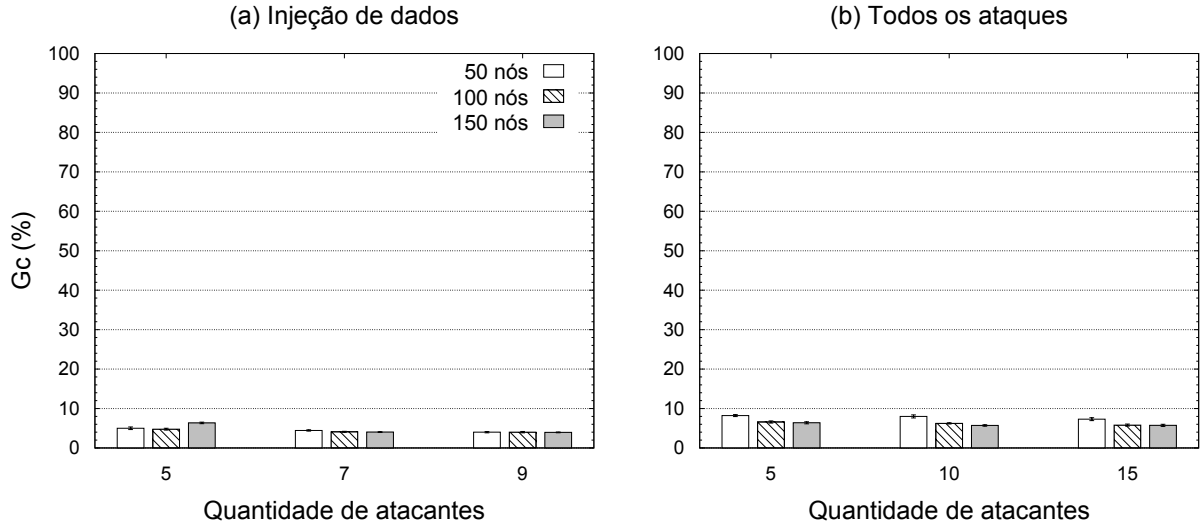


Figura 6.2: G_c em cenários urbanos sem o uso do QS^2

A Figura 6.3(a) apresenta os resultados obtidos pelo $PAN + QS^2$ diante de ataques de injeção de dados em um cenário urbano. Os resultados estão agrupados por quantidade de atacantes presente na rede. Nesses cenários, o G_c se apresenta inferior ao obtido com os cenários de validação. Isso pode ocorrer devido ao aumento da área de movimentação e do tempo de pausa e velocidade dos nós, que pode causar um atraso na propagação dos dados pela necessidade de recalcular rotas para os nós. Ainda, conforme observado anteriormente, a velocidade ajuda o QS^2 a obter informações sobre o comportamento dos demais nós, e o tempo de pausa prolongado dos cenários atuais pode causar um atraso no cálculo dessas informações.

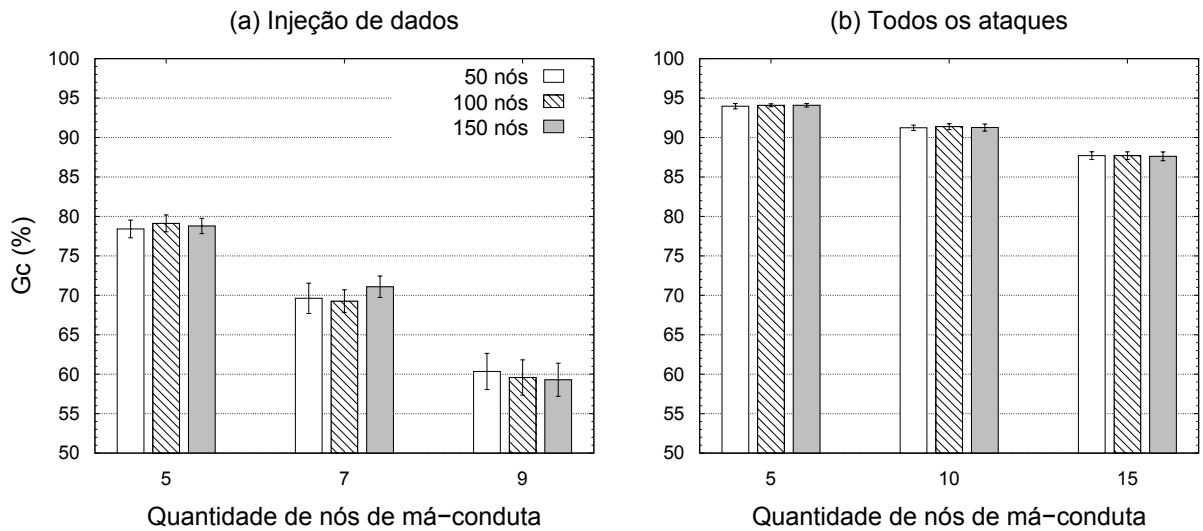


Figura 6.3: G_c em cenários urbanos

Também observa-se que o aumento da densidade da rede não varia em grande quantidade o G_c , enquanto considerado a mesma quantidade de atacantes. Os cenários com

todos os ataques, apresentados na Figura 6.3(b), também apresentam esse comportamento. Nesse cenário, o G_c é maior do que nos cenários com ataques de injeção de dados, e isso ocorre devido à mistura dos tipos de ataques. A quantidade total de atacantes é dividida entre todos os ataques, e os ataques de falta de cooperação e de temporização amenizam o impacto dos ataques de injeção de dados.

6.2.2 Eficiência

A eficiência de detecção dos ataques de injeção de dados, Figura 6.4(a), foi maior do que a obtida com os cenários de validação anteriores. Para os cenários urbanos, a tx_{det} é superior à 90%, e aumenta conforme a quantidade de nós de má-conduta também aumenta. Contudo, esse comportamento não se observa na detecção nos cenários com todos os ataques, conforme a Figura 6.4(b), em que a Tx_{det} se mantém entre 85% e 90%, independente da quantidade de nós de má-conduta ou da quantidade de nós na rede.

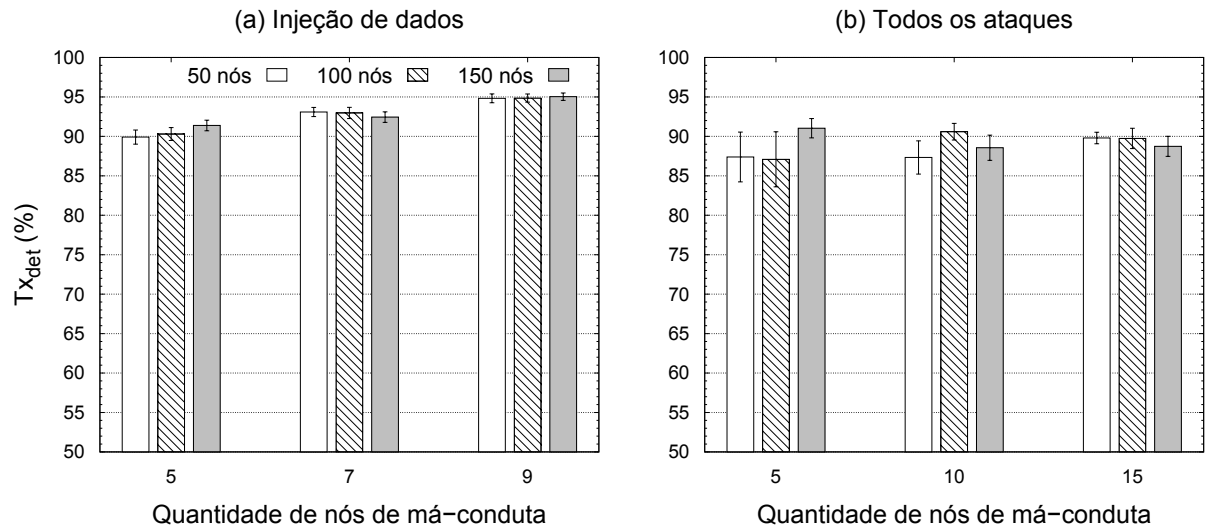


Figura 6.4: Tx_{det} em cenários urbanos

6.2.3 Dados falsos X dados desatualizados

Devido à discrepância entre o G_c mais baixo e a Tx_{det} maior, suspeitou-se que o motivo para a queda do G_c em tais cenários não seria a ação dos nós de má-conduta ou a perda de eficácia do QS^2 , e sim uma perda normal para o ambiente de rede, devido às suas características. Para verificar essa suposição, foram quantificadas as taxas de leituras que obtiveram dados falsos (Tx_{mis}) e dados desatualizados (Tx_{out}). Para ambos os ataques, a suposição se mostrou verdadeira.

Nos ataques de injeção de dados, a Tx_{mis} , Figura 6.5(a), é inferior a metade da Tx_{out} , Figura 6.5(b). Enquanto que a quantidade de dados falsos obtidos em leituras com 5 nós

de má-conduta é em média de 12%, a quantidade de leituras desatualizadas obtidas nesse mesmo cenário é de 39% em média. Enquanto a Tx_{mis} cresce em torno de 1% com o aumento da quantidade de nós no sistema, a Tx_{out} cresce em média 10% com o aumento da quantidade de nós.

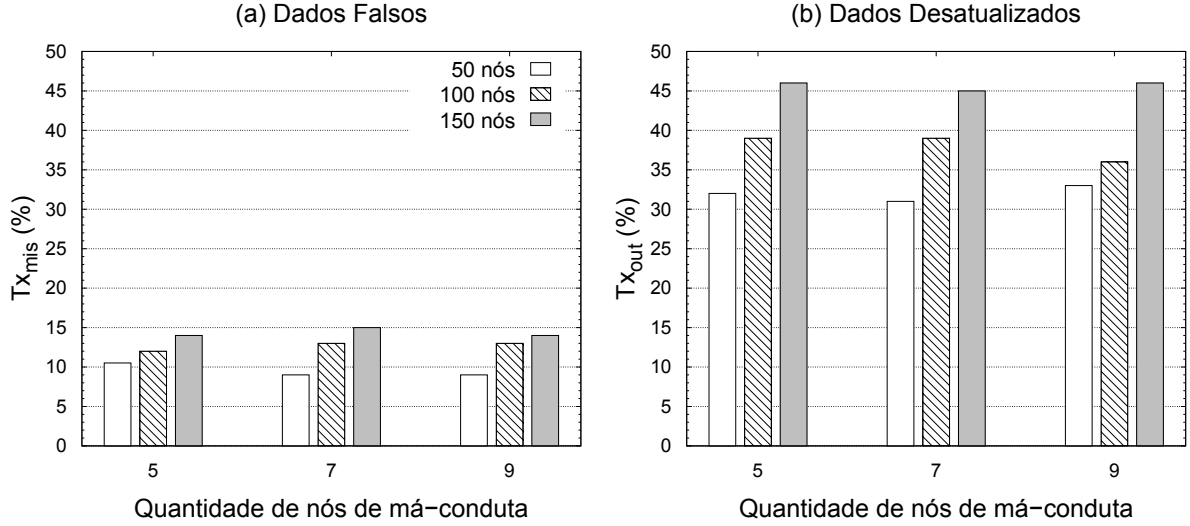


Figura 6.5: Tx_{mis} e Tx_{out} em cenários com ataque de injeção de dados

Esse comportamento se repete, porém em menor escala, nos cenários com todos os ataques, como mostram as Figuras 6.6(a) e 6.6(b). Isso evidencia que a queda do G_c nesses cenários, acontece em parte, devido à topologia da rede, como o tamanho da área de movimentação e a densidade de nós na rede. Ainda assim, o QS^2 obteve boas taxas de detecção e de G_c , sendo a última superior ao alcançado pelo PAN diante de ataques sem o uso do QS^2 nos cenários de validação.

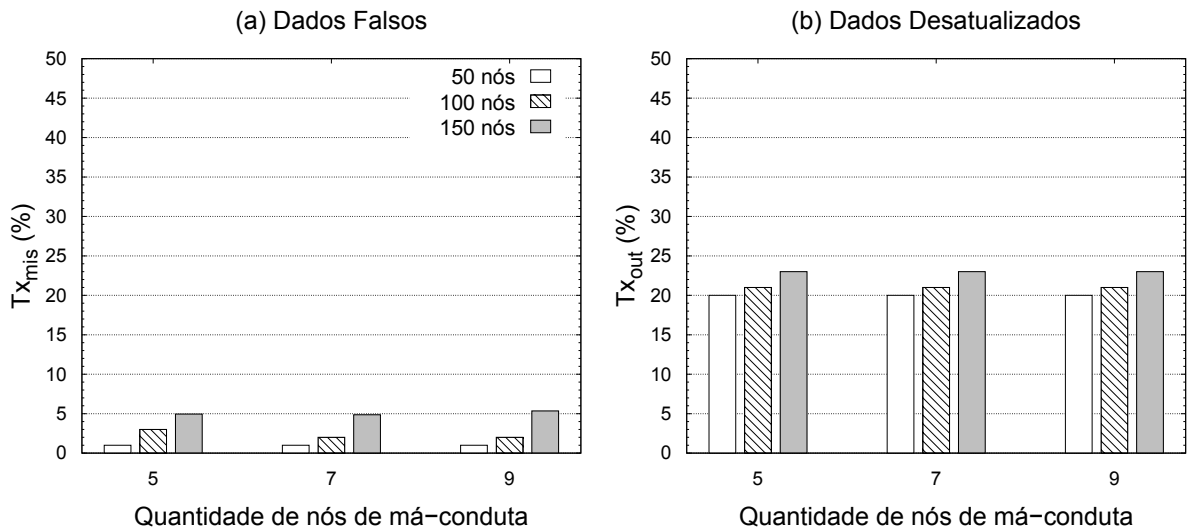


Figura 6.6: Tx_{mis} e Tx_{out} em cenários com todos os ataques

6.3 Ambiente de transporte - linhas de ônibus

Esse cenário baseia-se na implementação realizada em [73], aplicado em um ambiente de transporte suportado por uma arquitetura de roteamento chamada de *Ad Hoc City*. Ela é utilizada para a distribuição de informações entre veículos, usuários e terminais de transporte público e privado, sendo que os usuários podem participar dessa arquitetura utilizando dispositivos sem fio como *notebooks*, celulares ou *tablets*.

Nesse tipo de cenário, a mobilidade e a falta de infraestrutura são evidentes e precisam ser gerenciadas para que a aplicação de distribuição de mensagens para os veículos e pedestres possa funcionar de maneira adequada. A utilização dos sistemas de quórum para o apoio na tolerância a essas falhas também necessita ser robusta para garantir a entrega de informações íntegras e atualizadas para os usuários, evitando que nós de má-conduta prejudiquem a rede. Esses nós podem comprometer a funcionalidade da rede evitando que os nós confiáveis recebam dados úteis, ou ainda disseminar dados falsos que não servem aos usuários.

Esse cenário se caracteriza por um *backbone* composto por ônibus e veículos de entrega, que cobrem uma área específica na cidade. Esses veículos são organizados de maneira *ad hoc*, e fornecem acesso à rede para a comunicação em geral. Além disso, oito estações fixas são distribuídas ao longo da cidade, com o objetivo de melhorar a escalabilidade da rede e fornecer acesso à Internet aos usuários. O cenário corresponde ao sistema de ônibus (*King County Metro*) da cidade de *Seattle* (*Washington, USA*). O padrão de movimentação dos ônibus foi obtido por [73] através de observação da movimentação real dos ônibus durante os períodos da manhã e da tarde durante duas semanas (de 17 de novembro a 1 de dezembro de 2001). Um exemplo desse cenário é apresentado na Figura 6.7, em que os ônibus, representados por pontos cinza, e as estações fixas, apresentadas em branco, aparecem distribuídos sobre o mapa da cidade de *Seattle*.

Originalmente em [73], considera-se a existência de aproximadamente 850 ônibus movimentando-se em uma área de 5100km^2 . Devido a limitações computacionais, o cenário foi adaptado para uma topologia de 150 nós, distribuídos em uma área de 1500 x 2000 metros. Os nós seguem o padrão de movimentação dos ônibus, e movem-se entre 0 e 90 km/h. Foram utilizadas 8 estações fixas, distribuídas proporcionalmente no cenário, mantendo a arquitetura proposta por [73].

Devido às características da topologia da rede, alguns parâmetros referentes ao sistema de quórum PAN e à rede foram adaptados. O quórum de leitura (Q_r) é composto por quatro servidores, incluindo o agente, e o quórum de escrita (Q_w) é formado por todos os nós que recebem a escrita de um dado, sendo que cada nó dissemina os dados para quatro servidores a cada $T=200\text{ms}$. O conjunto de armazenamento (StS) é composto por 30 nós, escolhidos randomicamente. O intervalo de envio de escritas e leituras de cada nó é modelado seguindo a distribuição de Poisson, com $\lambda = 100$ para as escritas e $\lambda = 36$ para



Figura 6.7: Cenário de simulação de linhas de ônibus [69]

as leituras, e é dado em segundos. A taxa máxima de escritas é igual a $k_{envmax} = 0,018$ escritas por segundo, e a taxa de encaminhamento deve ser superior a $k_{encmin} = 0,15$ pacotes por segundo. Um resumo dos principais parâmetros empregados nas simulações desse cenário é apresentado na Tabela 6.2.

Tabela 6.2: Principais parâmetros de simulação dos cenários de transporte

Parâmetros	Valor
Quantidade de nós	150
Quantidade de nós no StS	30
Tempo de vida da rede	900 segundos
Velocidades máximas	0 a 90km/h
Raio de transmissão	106 metros
<i>Fanout</i> (F)	4 servidores
Quórum de leitura	4 servidores
Intervalo de propagação (T)	200ms e 3000ms
Quantidade de nós de má-conduta (f)	20%, 28%, 36%
Taxa de escrita (k_{envmax})	0,018 escritas por segundo
Taxa de encaminhamento (k_{encmin})	0,15 encaminhamentos por segundo

O padrão de movimentação utilizado refere-se a um intervalo de quinze minutos dos registros obtidos por [73]. Para simular cenários da rotina dos ônibus pela manhã, considerou-se os registros obtidos das 07:15 às 07:30, e para a tarde, utilizou-se os registros obtidos entre 17:15 e 17:30. Como protocolo de roteamento empregou-se o AODV e os resultados apresentados são a média de 35 simulações de 900 segundos, com um intervalo de confiança de 95%. Os cenários com ataques foram divididos em dois tipos: cenários

com 5, 7 e 9 nós de má-conduta, iniciando o ataque de injeção de dados nas operações de escrita, e cenários com 5, 10 e 15 nós iniciando todos os ataques em conjunto. Os ataques considerados no cenário com ataques em conjunto são: injeção de dados nas operações de escrita e leitura, falta de cooperação nas operações de escrita e de leitura e ataques de temporização ($T=3000\text{ms}$), sendo que cada ataque é desempenhado por 20% do total de nós de má-conduta considerados. As métricas de avaliação são o *grau de confiabilidade* (G_c) e a *taxa de detecção* (Tx_{det}), expressas nas Equações 5.1 e 5.2, respectivamente, e as taxas de *dados falsos* (Tx_{mis}) e de *dados atrasados* (Tx_{out}) obtidas pelas leituras nos quóruns, apresentadas nas Equações 5.5 e 5.6.

6.3.1 Grau de confiabilidade

Assim como nos cenários anteriores, a confiabilidade do sistema PAN sem o uso do QS^2 e diante de ataques é inferior a 10%. Isso se aplica aos cenários com ataques de injeção de dados (Figura 6.8(a)) e em cenários com todos os ataques (Figura 6.8(b)).

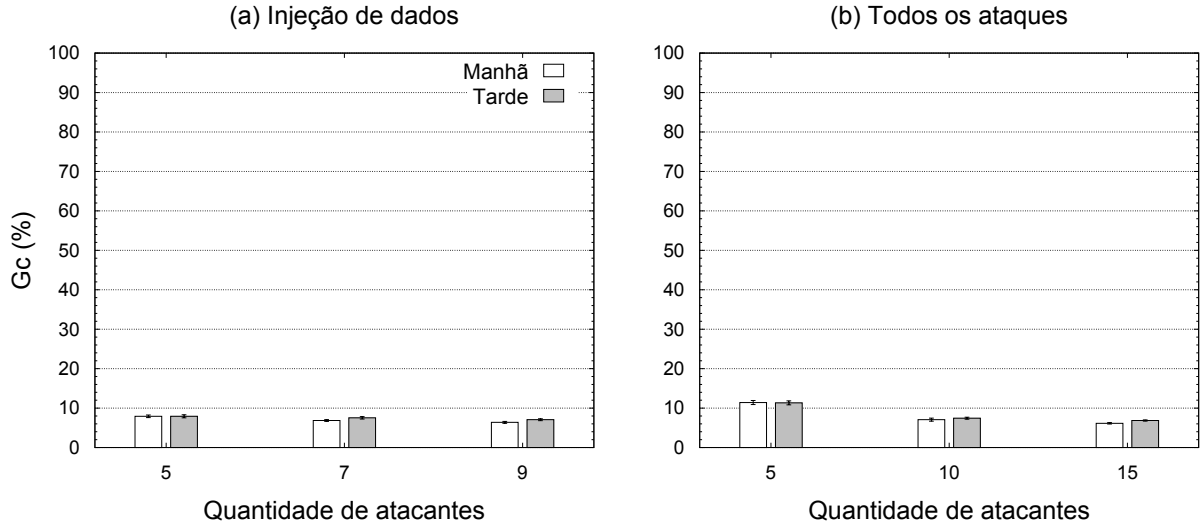


Figura 6.8: G_c em cenários de transporte sem o uso do QS^2

A Figura 6.9 mostra os resultados de G_c obtidos com os cenários de linhas de ônibus. Os resultados estão agrupados por quantidade de nós de má-conduta na rede, e são apresentados para os cenários da manhã (Figura 6.9(a)) e da tarde (Figura 6.9(b)), com ataques de injeção de dados e com todos os ataques em conjunto. Nesses ambientes, a confiabilidade alcançada com o $PAN + QS^2$ se apresenta acima de 60% para ambos os cenários da manhã e da tarde. Também observa-se que os cenários com ataques de injeção de dados e os cenários com todos os ataques se assemelham na confiabilidade obtida, contrariando os resultados obtidos com os cenários de validação, em que os cenários com todos os ataques apresentam resultados superiores a 92%. Isso ocorre devido ao padrão de movimentação dos nós, cujas velocidades variam de forma irregular. As diversas pa-

radas dos ônibus em pontos de embarque para usuários e terminais centrais ocasionam uma aceleração irregular, o que pode gerar perda dos pacotes de dados no meio sem fio. Além disso, o tempo de pausa desses veículos também tem uma grande variação, e como visto anteriormente, tempos de pausa prolongados não favorecem a disseminação e contabilização dos autoindutores por todos os nós da rede.

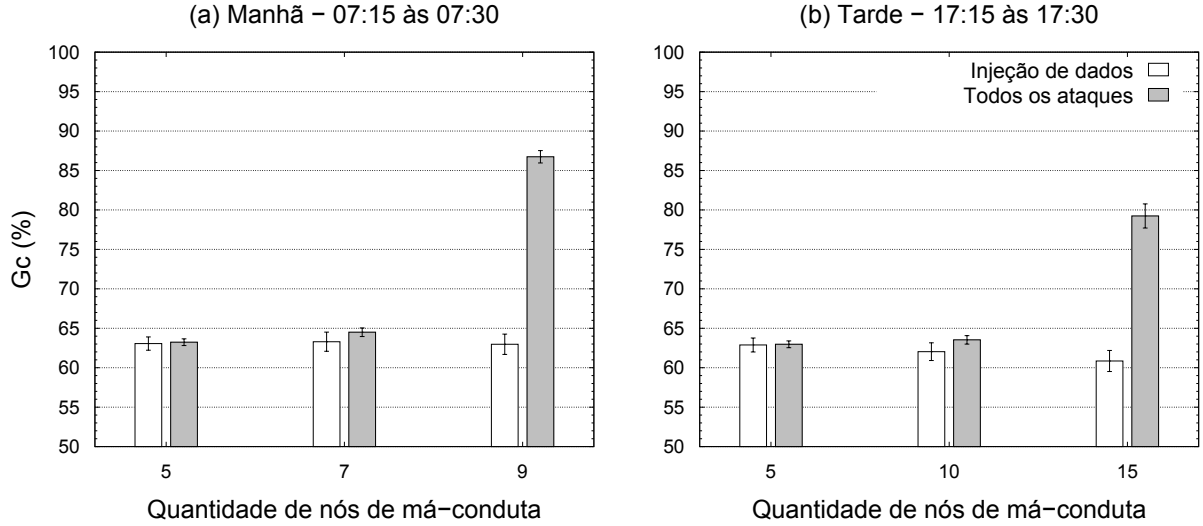


Figura 6.9: G_c em cenários de linhas de ônibus

6.3.2 Eficiência

A eficiência de detecção (Tx_{det}) do QS^2 nesses cenários, apresentada na Figura 6.10, reflete os resultados de G_c obtidos. Enquanto que a Tx_{det} para os cenários com ataques de injeção de dados, Figura 6.10(a), é superior a 65% e cresce conforme aumenta o número de nós de má-conduta na rede, a Tx_{det} para os cenários com todos os ataques, Figura 6.10(b), é inferior a 80%. A eficiência na detecção dos nós de má-conduta em cenários com todos os ataques é menor devido à falta de autoindutores que representam de uma forma adequada o ataque de temporização, e à demora para a detecção dos nós egoístas, que fazem parte dos nós de má-conduta presentes na rede.

Além disso, por conta do cenário utilizado, os nós podem encontrar dificuldades na troca de mensagens, e dessa forma, a detecção dos nós pode atrasar. Isso causa menores taxas de detecção dos nós de má-conduta.

6.3.3 Dados falsos X dados desatualizados

Da mesma forma que o cenário anterior, as taxas de dados falsos (Tx_{mis}) e de dados desatualizados (Tx_{out}) mostram que a quantidade de dados falsos obtidos pelos nós por meio de leituras no sistema de quórum é menor do que a quantidade de dados desatualizados. A

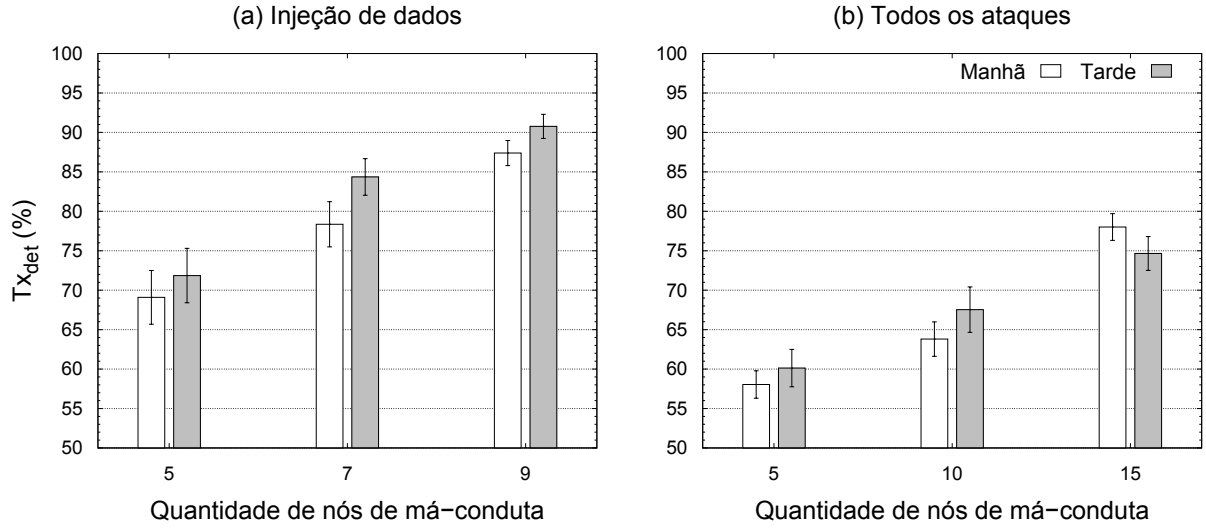


Figura 6.10: Tx_{det} em cenários de linhas de ônibus

Figura 6.11(a) apresenta as taxas Tx_{mis} e a Figura 6.11(b) apresenta os resultados para Tx_{out} , ambas para os ataques de injeção de dados. Nela, observa-se que em ambos os cenários, aproximadamente 50% dos dados obtidos pelos nós são descartados porque não estão atualizados, e apenas cerca de 10% são dados falsos, injetados por nós maliciosos.

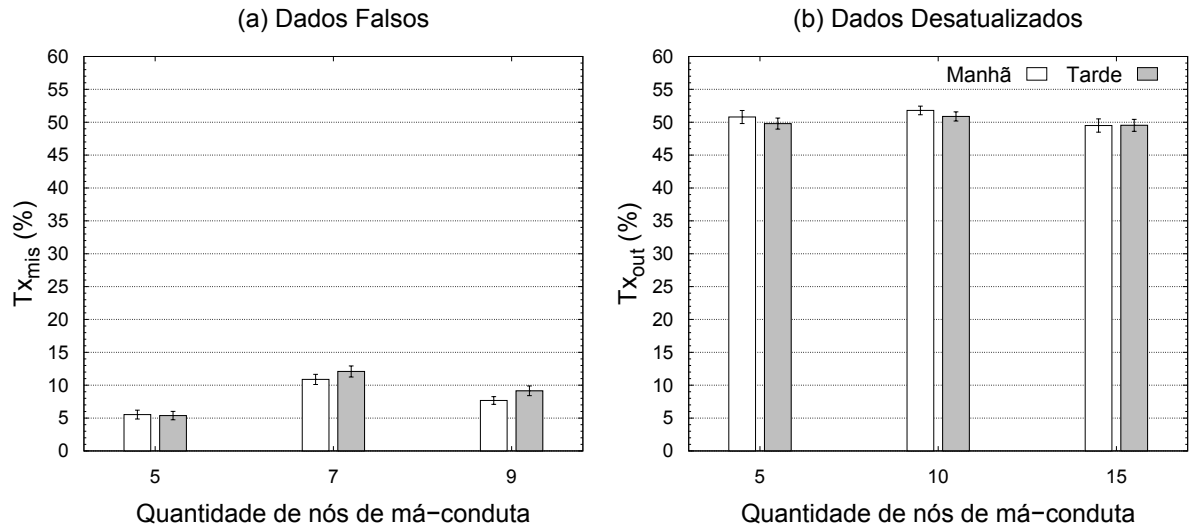


Figura 6.11: Tx_{mis} e Tx_{out} em cenários com ataque de injeção de dados

As taxas Tx_{mis} e Tx_{out} para os cenários com todos os ataques são apresentadas na Figura 6.12(a) e Figura 6.12(b), respectivamente. Esses cenários também apresentam um baixo percentual de dados falsos obtidos pelos nós, e mostram que a quantidade de dados desatualizados retornados por operações de leitura são maioria.

Tais resultados evidenciam que a rede apresenta dificuldade na entrega de dados, o que possivelmente é consequência do padrão de movimentação dos ônibus. Essa dificuldade em entregar dados, por sua vez, influencia na dificuldade de disseminação dos dados de

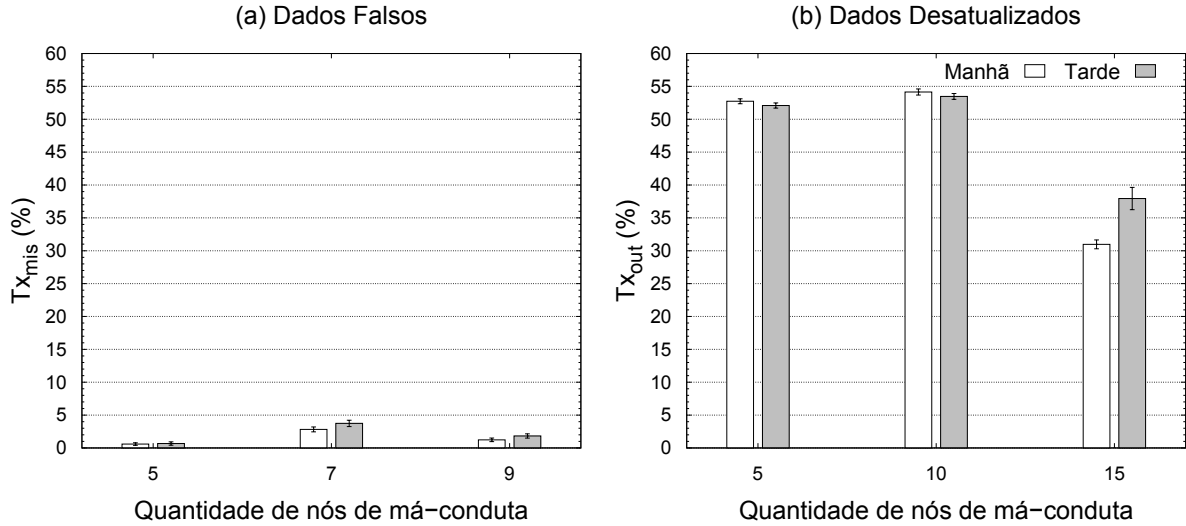


Figura 6.12: Tx_{mis} e Tx_{out} em cenários com todos os ataques

replicação e dos autoindutores utilizados pelo QS^2 para a detecção dos nós de má-conduta. A Tabela 5.4 resume os resultados do QS^2 diante de ambientes de MANETs reais.

Tabela 6.3: Síntese do uso do QS^2 em ambientes realísticos de MANETs

Métricas	Centro da cidade		Linhas de ônibus	
	Inj. de dados	Todos	Inj. de dados	Todos
G_c	70%	90%	70%	65%
Tx_{det}	92%	87%	80%	67%
Tx_{mis}	12%	4%	7%	3%
Tx_{out}	40%	22%	50%	45%

6.4 Resumo

Este capítulo apresentou a aplicação e a avaliação do QS^2 no apoio aos sistemas de quórum em cenários realísticos de MANETs. Foram empregados dois cenários realísticos: o primeiro é representado por um ambiente urbano de disseminação de informações comerciais no centro de uma cidade e o segundo é representado por um ambiente de transporte de disseminação de informações sobre a situação do tráfego e do horário de linhas de ônibus. Nesses cenários, o QS^2 se mostrou efetivo na identificação de nós de má-conduta e na melhoria da confiabilidade dos dados replicados. Porém, o correto funcionamento do QS^2 depende das condições da rede, como mobilidade e entrega de mensagens para alcançar um padrão aceitável de taxa de dados corretos e atualizados no sistema de replicação.

CAPÍTULO 7

CONSIDERAÇÕES FINAIS

As MANETs são redes sem fio formadas dinamicamente que possuem como característica a descentralização das operações, a mobilidade e a escassez de recursos. Essas características podem tornar os serviços e as aplicações indisponíveis ou resultar em informações desatualizadas, devido à mobilidade ou à falta de energia dos nós. Os serviços de operação de rede, tais como serviços de localização de recursos e gerência de mobilidade, apoiam o funcionamento das aplicações através do envio e da gerência de informações. Desta forma, tais serviços precisam ser robustos, necessitando de garantias de disponibilidade e de confiabilidade dos seus dados.

Uma das formas comumente empregadas para tolerar falhas é por meio da redundância das informações, obtida através de técnicas de replicação dos dados. Porém, as abordagens clássicas de replicação de dados, quando aplicadas em MANETs, apresentam um alto custo. Isso ocorre devido às restrições impostas por essas abordagens, que não refletem as características do ambiente das MANETs. Os sistemas de quórum são mecanismos eficientes para a replicação de dados, pois distribuem a carga de leituras e de escritas entre vários servidores e diminuem o custo de comunicação replicando os dados em um subconjunto de servidores. Contudo, esses sistemas são vulneráveis à ação de nós de má-conduta nas MANETs, que têm como objetivo a negação do serviço da rede. Este trabalho avaliou por meio de simulações um sistema de quórum probabilístico para MANETs diante dos ataques de falta de cooperação, de temporização e de injeção de dados. Verificou-se que os nós de má-conduta diminuem a confiabilidade da replicação. Dessa forma, esses sistemas necessitam de uma forma de tolerar a presença de nós de má-conduta nas operações de replicação, garantindo a continuidade dos serviços.

Nesse contexto, este trabalho propôs um esquema de exclusão de nós egoístas e maliciosos das operações de replicação de um sistema de quórum para MANETs, com o objetivo de tolerar a presença desses nós e minimizar sua participação nas operações de replicação dos sistemas de quórum. Esse esquema, chamado de QS^2 , é inspirado nos mecanismos de sensoramento em quórum e seleção por parentesco, ambos encontrados em bactérias. O QS^2 identifica os nós de má-conduta através do seu comportamento e evita a escolha de tais nós para a participação nos quóruns. Esse esquema é autônomo e auto-organizado, sendo que os nós não trocam informações de reputação com outros nós e baseiam a detecção na sua própria experiência com os demais nós da rede. O QS^2 também utiliza a própria troca de mensagens de escrita para a detecção dos nós de má-conduta, o que não gera maiores custos de comunicação para os nós da rede.

As operações de escrita e de encaminhamentos de dados do sistema de quórum são utilizadas pelo QS^2 como um indicativo do bom ou mau comportamento dos nós. A quantidade de escritas e encaminhamentos são contabilizadas e comparadas à um padrão considerado aceitável para cada um deles. Os nós que apresentam uma quantidade de escritas enviadas superior ao permitido são considerados maliciosos, e aqueles que apresentam uma quantidade de encaminhamentos de dados inferior ao indicado são considerados nós egoístas. Cada nó monitora o envio de escritas e encaminhamentos pelos demais nós, e dá preferência em realizar as operações de leitura e de escrita do sistema de quórum com aqueles nós que apresentam taxas dentro do limite definido. O esquema QS^2 enfatiza as características de autonomia, auto-organização e utilização de poucos recursos, que por sua vez implicam em alguns obstáculos para a solução, tais como a detecção errônea de nós confiáveis, escalabilidade inadequada para determinados sistemas e um custo de processamento pelo uso de um esquema de assinaturas.

O esquema QS^2 foi implementado e adicionado ao PAN, um sistema de quórum probabilístico para MANETs, e avaliado na presença de nós de má-conduta na forma de ataques de falta de cooperação, temporização e injeção de dados. Os resultados obtidos mostram que o QS^2 aumentou a confiabilidade de um sistema de quórum probabilístico para MANETs em até 87%, como nos cenários com ataques de injeção de dados nas operações de escrita. Além disso, o QS^2 apresentou uma grande eficácia na detecção de nós egoístas e maliciosos, com uma baixa taxa de falsos positivos. Já para os ataques de temporização, o QS^2 apresentou uma leve melhora na confiabilidade dos dados, sendo que em alguns cenários houve uma pequena perda, em torno de 1%.

O QS^2 também foi avaliado no apoio aos sistemas de quórum em dois cenários realísticos: na distribuição de informações do comércio local no centro de uma cidade e na distribuição de informações de tráfego e horário de ônibus em linhas de ônibus. Nesses cenários, o QS^2 se mostrou efetivo na detecção de nós de má-conduta, sendo que a confiabilidade dos dados obteve um aumento entre 50% e 85% em relação ao mesmo cenário sem o uso do QS^2 . Verificou-se ainda que a quantidade de dados desatualizados retornados em operações de leitura é muito maior do que a quantidade de dados falsos nos nós.

Como trabalhos futuros, pretende-se avaliar o comportamento de diversos serviços de operação da rede e caracterizar o tráfego de dados desses serviços. Além disso, pretende-se avaliar o QS^2 com o uso desses dados coletados, e analisar o impacto do uso de limites de escritas e encaminhamentos adaptáveis ao ambiente. Pretende-se também verificar e analisar a identificação e o uso de autoindutores que melhor caracterizem os ataques de falta de cooperação e de temporização nos sistemas de quórum, e dessa forma, permitir que o sistema identifique e exclua com maior precisão esses nós. Outras intenções de trabalhos futuros são a análise do custo dos modelos de assinatura utilizados pelo QS^2 , a verificação do uso do QS^2 em outros modelos de replicação e a análise do impacto do comportamento intermitente dos nós de má-conduta.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Marco Conti, Song Chong, Serge Fdida, Weijia Jia, Holger Karl, Ying-Dar Lin, Petri Mähönen, Martin Maier, Refik Molva, Steve Uhlig, and Moshe Zukerman. Research challenges towards the future internet. *Computer Communications*, 34(18):2115 – 2134, 2011.
- [2] Chi Zhang, Yang Song, and Yuguang Fang. Modeling secure connectivity of self-organized wireless ad hoc networks. In *Proceedings of the 27th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '08)*, Los Alamitos, CA, USA, 2008. IEEE Communications Society.
- [3] Abdelouahid Derhab and Nadjib Badache. Data replication protocols for mobile ad-hoc networks: a survey and taxonomy. *IEEE Communications Surveys and Tutorials*, 11:33–51, 2009.
- [4] Abdelsalam A. Helal, Bharat K. Bhargava, and Abdelsalam A. Heddaya. *Replication Techniques in Distributed Systems*. Kluwer Academic Publishers, Norwell, MA, USA, 1996.
- [5] Rashedur M. Rahman, Ken Barker, and Reda Alhajj. Study of different replica placement and maintenance strategies in data grid. In *Proceedings of the 7th IEEE International Symposium on Cluster Computing and the Grid (CCGRID '07)*, pages 171–178, Washington, DC, USA, 2007.
- [6] Navin Budhiraja, Keith Marzullo, Fred B. Schneider, and Sam Toueg. *The primary-backup approach*, chapter 8, pages 199–216. Addison-Wesley, New York, NY, USA, 2 edition, 1993.
- [7] Fred B. Schneider. *Replication management using the state-machine approach*, pages 169–197. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1993.
- [8] Dahlia Malkhi and Michael Reiter. Byzantine quorum systems. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC '97)*, pages 569–578, New York, NY, USA, 1997. ACM.
- [9] Jun Luo, Jean-Pierre Hubaux, and Patrick Th. Eugster. PAN: Providing reliable storage in mobile ad hoc networks with probabilistic quorum systems. In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '03)*, pages 1–12, New York, NY, USA, 2003. ACM.

- [10] Daniela Tulone. Ensuring strong data guarantees in highly mobile ad hoc networks via quorum systems. *Ad Hoc Networks*, 5(8):1251–1271, 2007.
- [11] Dahlia Malkhi, Michael K. Reiter, Avishai Wool, and Rebecca N. Wright. Probabilistic quorum systems. *Information Computing*, 170(2):184–206, 2001.
- [12] Mansoor Alicherry, Angelos D. Keromytis, and Angelos Stavrou. Evaluating a collaborative defense architecture for manets. In *Proceedings of the 3rd IEEE International Conference on Internet Multimedia Services Architecture and Applications*, IMSAA’09, pages 229–234, Piscataway, NJ, USA, 2009. IEEE Press.
- [13] Marco Conti, Enrico Gregori, and Gaia Maselli. Cooperation issues in mobile ad hoc networks. In *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops*, ICDCSW ’04, pages 803–808, Washington, DC, USA, 2004.
- [14] Yair Amir, Brian A. Coan, Jonathan Kirsch, and John Lane. Byzantine replication under attack. In *Proceedings of the 38th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN ’08)*, pages 197–206, Washington, DC, 2008. IEEE Computer Society.
- [15] Gregory Chockler, Seth Gilbert, and Boaz Patt-Shamir. Communication-efficient probabilistic quorum systems for sensor networks. In *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom ’06)*, pages –117, Março 2006.
- [16] Elisa Mannes, Eduardo da Silva, Michele Nogueira, and Aldri L. dos Santos. Implications of misbehaving attacks on probabilistic quorum systems for manets. In *Proceedings of the International Conference on Security and Cryptography (SECRYPT)*, pages 189–195, Athens, Greece, Julho 2010.
- [17] Jonathan Kirsch and Yair Amir. Paxos for system builders. In *Proceedings of the 2nd Workshop on Large-Scale Distributed Systems and Middleware (LADIS ’08)*, pages 1–6, Yorktown Heights, New York, USA, 2008. ACM.
- [18] Seth Gilbert. Virtual infrastructure for wireless ad hoc networks, 2007.
- [19] Sonja Buchegger and Jean-Yves Le Boudec. Self-policing mobile ad hoc networks by reputation systems. *Communications Magazine, IEEE*, 43(7):101–107, Julho 2005.
- [20] Sonja Buchegger and Jean-Yves Le Boudec. Performance analysis of the confidant protocol. In *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking and computing (MobiHoc ’02)*, pages 226–236. ACM, 2002.

- [21] Hao Yang, Xiaoqiao Meng, and Songwu Lu. Self-organized network-layer security in mobile ad hoc networks. In *Proceedings of the 1st ACM workshop on Wireless security (WiSE '02)*, pages 11–20. ACM, 2002.
- [22] Natalia Castro Fernandes, Marcelo Duffles Donato Moreira, and Otto Carlos Muniz Bandeira Duarte. A self-organized mechanism for thwarting malicious access in ad hoc networks. In *Proceedings of the 29th conference on Information communications (INFOCOM'10)*, pages 266–270. IEEE Press, 2010.
- [23] Sonja Buchegger and Jean-Yves Le Boudec. A robust reputation system for p2p and mobile ad-hoc networks. In *3th Workshop on the Economics of Peer-to-Peer Systems (P2PEcon '04)*, 2004.
- [24] S. Balasubramaniam, K. Leibnitz, P. Lio, D. Botvich, and M. Murata. Biological principles for future internet architecture design. *Communications Magazine, IEEE*, 49(7):44–52, july 2011.
- [25] Divyakant Agrawal and Amr El Abbadi. An efficient and fault-tolerant solution for distributed mutual exclusion. *ACM Transactions on Computing Systems*, 9:1–20, Fevereiro 1991.
- [26] Mamoru Maekawa. A n algorithm for mutual exclusion in decentralized systems. *ACM Transactions on Computing Systems*, 3:145–159, Maio 1985.
- [27] Rida A. Bazzi. Synchronous byzantine quorum systems. *Distributed Computing*, 13(1):45–52, 2000.
- [28] Alysson Neves Bessani, Joni da Silva Fraga, and Lau Cheuk Lung. BTS: a byzantine fault-tolerant tuple space. In *Proceedings of the 21st ACM Symposium on Applied Computing (SAC '06)*, pages 429–433, New York, NY, USA, 2006. ACM.
- [29] Lorenzo Alvisi, Evelyn Tumlin Pierce, Dahlia Malkhi, Michael K. Reiter, and Rebecca N. Wright. Dynamic byzantine quorum systems. In *Proceedings of the 2000 International Conference on Dependable Systems and Networks (DSN '00)*, pages 283–292, Washington, DC, USA, 2000. IEEE Computer Society.
- [30] Gregory Chockler, Seth Gilbert, Vincent Gramoli, Peter M. Musial, and Alex A. Shvartsman. Reconfigurable distributed storage for dynamic networks. *Journal Parallel Distributed Computing*, 69:100–116, Janeiro 2009.
- [31] Sangeeta Bhattacharya. Randomized location service in mobile ad hoc networks. In *Proceedings of the 6th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems, MSWIM '03*, pages 66–73, New York, NY, USA, 2003. ACM.

- [32] Roy Friedman, Gabriel Kliot, and Chen Avin. Probabilistic quorum systems in wireless ad hoc networks. In *International Conference on Dependable Systems and Networks (DSN '08)*, pages 277–286, Washington, DC, 2008. IEEE Computer Society.
- [33] Vincent Gramoli and Michel Raynal. Timed quorum systems for large-scale and dynamic environments. *Principles of Distributed Systems*, pages 429–442, 2007.
- [34] Jean-Philippe Martin, Lorenzo Alvisi, and Michael Dahlin. Small byzantine quorum systems. In *Proceedings of the 2002 International Conference on Dependable Systems and Networks (DSN '02)*, pages 374–388, Washington, DC, USA, 2002. IEEE Computer Society.
- [35] Moni Naor and Udi Wieder. Scalable and dynamic quorum systems. *Distributed Computing*, 17:311–322, Maio 2005.
- [36] Ittai Abraham and Dahlia Malkhi. Probabilistic quorums for dynamic systems. *Distributed Computing*, 18(2):113–124, 2005.
- [37] Daniela Tulone and Erik D. Demaine. Revising quorum systems for energy conservation in sensor networks. In *Proceedings of the International Conference on Wireless Algorithms, Systems and Applications (WASA '07)*, pages 147–157, Washington, DC, USA, 2007. IEEE Computer Society.
- [38] Dahlia Malkhi, Michael Reiter, Avishai Wool, and Rebecca N. Wright. Probabilistic byzantine quorum systems. In *Proceedings of the 17th Annual ACM Symposium on Principles of Distributed Computing (PODC '98)*, page 321, New York, NY, USA, 1998. ACM.
- [39] Patrick T. Eugster, Rachid Guerraoui, Anne-Marie Kermarrec, and Laurent Massoulié. Epidemic information dissemination in distributed systems. *Computer*, 37(5):60–67, 2004.
- [40] Marcos D. Ortiz, Acélio S. C. Aguiar, Diogo A. Lima, Marcial Fernandez, and José N. de SOUSA. Análise, implementação e teste de uma estratégia autônoma de incentivo à cooperação em redes ad-hoc. In *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC '07)*, pages 235 – 248, Belém, PA, 2007.
- [41] Jiangyi Hu and Mike Burmester. Cooperation in mobile ad hoc networks. In *Computer Communications and Networks*, pages 1–15. Springer London, 2009.
- [42] Yi-an Huang and Wenke Lee. A cooperative intrusion detection system for ad hoc networks. In *Proceedings of the 1st ACM Workshop on Security of Ad hoc and Sensor Networks (SASN '03)*, pages 135–147, New York, NY, USA, 2003. ACM.

- [43] Vinod Shukla and Daji Qiao. Distinguishing data transience from false injection in sensor networks. In *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '07)*, pages 41–50, 2007.
- [44] Elisa Mannes, Eduardo da Silva, and Aldri L. dos Santos. Analisando o desempenho de um sistema de quóruns probabilístico para manets diante de ataques maliciosos. In *Anais do IX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg '09)*, pages 71–84, Setembro 2009.
- [45] Theodore S. Rappaport. *Comunicação sem fio: princípios e práticas*. Pearson, São Paulo, SP, Brasil, 2 edition, Dezembro 2008.
- [46] Andres Rojas, Philip Branch, and Grenville Armitage. Experimental validation of the random waypoint mobility model through a real world mobility trace for large geographical areas. In *Proceedings of the 8th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWiM '05)*, pages 174–177, New York, NY, USA, 2005. ACM.
- [47] Emmanouil A. Panaousis, Levon Nazaryan, and Christos Politis. Securing aodv against wormhole attacks in emergency manet multimedia communications. In *Proceedings of the 5th International Mobile Multimedia Communications Conference (MobiMedia '09)*, pages 34:1–34:7, 2009.
- [48] Madjid Merabti Sohail Abbas and David Llewellyn-Jones. A survey of reputation based schemes for manet. In *11th Annual Conference on the Convergence of Telecommunications, Networking and Broadcasting, PGNet '10*, pages 233–238, 2010.
- [49] Jun Cheol Park and Sneha Kumar Kasera. Securing ad hoc wireless networks against data injection attacks using firewalls. In *Wireless Communications and Networking Conference, 2007.WCNC 2007. IEEE*, pages 2843–2848, Março 2007.
- [50] Zhengjian Zhu, Qingping Tan, and Peidong Zhu. An effective secure routing for false data injection attack in wireless sensor network. In *Managing Next Generation Networks and Services*, volume 4773 of *Lecture Notes in Computer Science*, pages 457–465. Springer Berlin / Heidelberg, 2007.
- [51] Sencun Zhu, Sanjeev Setia, Sushil Jajodia, and Peng Ning. Interleaved hop-by-hop authentication against false data injection attacks in sensor networks. *ACM Transactions in Sensor Networks*, 3, Agosto 2007.
- [52] Falko Dressler and Ozgur B. Akan. A survey on bio-inspired networking. *Computing Network*, 54:881–900, 2010.

- [53] Andy M. Tirrel, Eduardo Sanchez, Dario Floreano, Gianluca Tempesti, Daniel Mange, Juan-Manuel MÓreno, Jay Roenberg, and Alessandro EP Villa. Poetic tissue: An integrated architecture for bio-inspired hardware. *Proceeding of the 5th International Conference on Evolvable Systems: From Biology to Hardware (ICES '03)*, pages 129–140, 2003.
- [54] Gianni Di Caro, Frederick Ducatelle, and Luca Maria Gambardella. Anthocnet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications*, 16:443–455, 2005.
- [55] Shahab Kamali and Jaroslav Opatrny. A hybrid ant-colony routing algorithm for mobile ad-hoc networks. In *Complex Sciences*, volume 5 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 1337–1354. Springer Berlin Heidelberg, 2009.
- [56] Jean-Yves Le Boudec and Slavisa Sarafijanovic. An artificial immune system approach to misbehavior detection in mobile ad hoc networks. In *Biologically Inspired Approaches to Advanced Information Technology*, volume 3141 of *Lecture Notes in Computer Science*, pages 396–411. Springer Berlin / Heidelberg, 2004.
- [57] Alexander Tyrrell, Gunther Auer, and Christian Bettstetter. Fireflies as role models for synchronization in ad hoc networks. In *Proceedings of the 1st International Conference on inspired Models of Network, Information and Computing Systems (BI-ONETICS '06)*, New York, NY, USA, 2006. ACM.
- [58] Vasileios Pappas, Dinesh Verma, Bong-Jun Ko, and Ananthram Swami. A circulatory system approach for wireless sensor networks. *Ad Hoc Networks*, 7:706–724, 2009.
- [59] Wai-Leung L. Ng and Bonnie L. Bassler. Bacterial quorum-sensing network architectures. *Annual Review of Genetics*, 43(1):197–222, 2009.
- [60] Tamás Czárán and Rolf F. Hoekstra. Microbial communication, cooperation and cheating: Quorum sensing drives the evolution of cooperation in bacteria. *Public Library of Science ONE*, 4(8):6655, Agosto 2009.
- [61] William Donald Hamilton. The genetical evolution of social behaviour. *Journal of Theoretical Biology*, 7(1):1 – 16, 1964.
- [62] Jack Dockery and James Keener. A mathematical model for quorum sensing in *pseudomonas aeruginosa*. *Bulletin of Mathematical Biology*, 63:95–116, 2001.
- [63] John P Ward, James R King, Adrian J Koerber, Paul Williams, J M Croft, and Elizabeth Sockett. Mathematical modelling of quorum sensing in bacteria. *Journal Of Mathematics Applied In Medicine And Biology*, 18(3):263–292, 2001.

- [64] Melissa B. Miller and Bonnie L. Bassler. Quorum sensing in bacteria. *Annual Review of Microbiology*, 55:165–199, 2001.
- [65] Steve P. Diggle, Stuart A. West, Andy. Gardner, and Ashleigh S. Griffin. Communication in bacteria. *Sociobiology of Communication: an interdisciplinary perspective*, pages 11–31, 2008.
- [66] Shidi Xu, Yi Mu, and Willy Susilo. Efficient authentication scheme for routing in mobile ad hoc networks. In *Proceedings of the Embedded and Ubiquitous Computing Workshop (EUC '05)*, volume 3823, pages 854–863. Springer Berlin, 2005.
- [67] Rehan Akbani, Turgay Korkmaz, and G. V. S. Raju. Heap: A packet authentication scheme for mobile ad hoc networks. *Ad Hoc Networking*, 6:1134–1150, September 2008.
- [68] Hongwei Li. A hierarchical identity-based encryption for manets. *Proceedings of the International Conference on Computational Problem-Solving (ICCP '11)*, pages 330 – 333, October 2011.
- [69] Elisa Mannes, Michele Nogueira, and Aldri dos Santos. Um esquema bio-inspirado para tolerância à má-conduta em sistemas de quórum para manets. In *Anais do XI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg '11)*, pages 239–252, Brasília, DF, Novembro 2011.
- [70] Elisa Mannes, Michele Nogueira, and Aldri dos Santos. A bio-inspired scheme on quorum systems for reliable services data management in manets. In *IEEE/IFIP Network Operations and Management Symposium (NOMS '12)*, Hawaii, USA, Abril 2012. (aceito para publicação).
- [71] Christian Becker, Martin Bauer, and Jörg Hähner. Usenet-on-the-fly: supporting locality of information in spontaneous networking environments. In *Workshop on Ad Hoc Communications and Collaboration in Ubiquitous Computing Environments*. ACM Press, 2002.
- [72] Jing Tian, Joerg Haehner, Christian Becker, Illya Stepanov, and Kurt Rothermel. Graph-based mobility model for mobile ad hoc network simulation. In *Proceedings of the 35th Annual Simulation Symposium*, pages 337–, Washington, DC, USA, 2002. IEEE Computer Society.
- [73] Jorjeta G. Jetcheva, Yih-Chun Hu, Santashil PalChaudhuri, Amit K. Saha, and David B. Johnson. Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture. In *Proceedings of the 5th IEEE Workshop on Mobile Computing Systems and Applications*, pages 32–43, 2003.

ANEXO

As contribuições obtidas pela execução deste trabalho puderam ser publicadas em conferências nacionais e internacionais, listadas abaixo.

1. MANNES, Elisa; NOGUEIRA, Michele; SANTOS, Aldri L. Serviços Confiáveis em MANETs Baseado em Sistema de Quórum Tolerante a Má-conduta. In: Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2012, Ouro Preto. Anais do XXX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2012 (aceito para publicação)
2. MANNES, Elisa; NOGUEIRA, Michele; SANTOS, Aldri L. A Bio-inspired scheme on quorum systems for reliable services data management in MANETs. In: IEEE/IFIP Network Operations and Management Symposium (NOMS), 2012, Hawaii, Estados Unidos, 2012 (aceito para publicação)
3. MANNES, Elisa; NOGUEIRA, Michele; SANTOS, Aldri L. Um esquema bio-inspirado para Tolerância à Má-conduta em Sistemas de Quórum para MANETs. In: XI Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg), 2011, Brasília. Anais do XI Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, 2011. p. 239-252.
4. MANNES, Elisa; SILVA, Eduardo da; NOGUEIRA, Michele; SANTOS, Aldri L. Implications of Misbehaving Attacks on Probabilistic Quorum System for MANETs. In: International Conference on Security and Cryptography (SECRYPT), 2010, Atenas. Proceedings of the International Conference on Security and Cryptography, 2010. p. 189-195.
5. MANNES, Elisa; SILVA, Eduardo da; SANTOS, Aldri L. Analisando o desempenho de um sistema de quóruns probabilístico para MANETs diante de ataques maliciosos. In: IX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg), 2009, Campinas. Anais do IX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais, 2009. p. 71-84.